



Artikel Penelitian

Optimasi RoBERTa dengan Hyperparameter Tuning untuk Deteksi Emosi berbasis Teks

Elvanro Marthen Pusung^a, Ika Novita Dewi^{a}*^aProgram Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang, 50131

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 11 November 2024

Revisi Akhir: 14 Januari 2025

Diterbitkan Online: 01 Februari 2025

KATA KUNCI

Deteksi Emosi,
RoBERTa,
Hyperparameter Tuning,
Teks Bahasa Indonesia,
Natural Language Processing

KORESPONDENSI

E-mail: ikadewi@dsn.dinus.ac.id*

ABSTRACT

Deteksi emosi berbasis teks menjadi salah satu topik utama pembahasan dalam pemrosesan bahasa alami. Hal ini dapat disebabkan karena adanya peningkatan jumlah data yang dihasilkan dari interaksi digital, seperti media sosial dan aplikasi pesan instan. Deteksi emosi bertujuan untuk mengenali dan menganalisis respons emosional individu dalam situasi tertentu, dengan penerapan di berbagai bidang seperti interaksi manusia dengan komputer, analisis sentimen, dan layanan kesehatan mental. Penelitian ini bertujuan untuk meningkatkan performa deteksi emosi berbasis teks Bahasa Indonesia menggunakan model RoBERTa. RoBERTa merupakan salah satu varian dari model BERT (*Bidirectional Encoder Representations from Transformers*) yang telah dioptimalkan untuk pemrosesan bahasa alami dengan mempertimbangkan konteks linguistik secara lebih mendalam. Implementasi RoBERTa dalam deteksi emosi akan dioptimalkan melalui *hyperparameter tuning* seperti Grid Search, Randomized Search, dan Bayesian Optimization untuk memperoleh kombinasi nilai learning rate, batch size, dan epoch yang optimal, dengan tujuan meningkatkan akurasi dan kemampuan model dalam mendeteksi emosi secara konsisten. Deteksi emosi menggunakan dataset yang terdiri dari 7080 data *tweet* publik dengan enam kategori emosi, yaitu *anger*, *fear*, *joy*, *love*, *sad*, dan *neutral*. Hasil eksperimen menunjukkan bahwa metode Bayesian Optimization memberikan kombinasi hyperparameter yang optimal dan meningkatkan performa deteksi emosi pada teks berbahasa Indonesia dengan hasil akurasi sebesar 83,64% dan nilai precision, recall dan F1-score secara berurutan adalah 83,75%, 83,64%, dan 83,58%.

1. PENDAHULUAN

Deteksi emosi berbasis teks telah menjadi topik penting dalam bidang pengolahan bahasa alami (*Natural Language Processing* - NLP), terutama dengan meningkatnya interaksi pengguna di platform digital yang menghasilkan data teks dalam jumlah besar [1], [2]. Deteksi emosi merupakan proses identifikasi dan klasifikasi keadaan emosional individu melalui berbagai sinyal, seperti teks, suara, ekspresi wajah, atau sinyal fisiologis[3]. Tujuan utama deteksi emosi adalah memahami dan menganalisis respons emosional seseorang dalam konteks tertentu, yang dapat diterapkan dalam berbagai bidang seperti interaksi manusia-komputer, analisis sentimen, dan kesehatan mental[4]. Saat ini terdapat beberapa penelitian yang membahas deteksi emosi berdasarkan teks. Misalnya, penelitian [5] menerapkan 1D-CNN

untuk mendeteksi sentimen positif, negatif, dan netral pada ulasan produk Emina. Lalu, penelitian [6] menggunakan metode *Deep Neural Network* untuk mendeteksi sentimen positif, negatif, dan netral terhadap perusahaan publik. Emosi merupakan kondisi psikologis yang terdiri dari tiga komponen berbeda, yaitu pengalaman subjektif, respons fisiologis, serta komponen ekspresif atau perilaku[7]. Selain itu, emosi dapat diartikan sebagai respons tubuh terhadap situasi tertentu, sering kali terkait dengan proses berpikir (kognitif) seseorang, di mana jenis dan intensitas emosi dipengaruhi oleh persepsi individu terhadap situasi yang dihadapi[7].

Model RoBERTa, varian yang diperkuat dari BERT (*Bidirectional Encoder Representations*), telah dioptimalkan untuk berbagai bahasa, termasuk adaptasinya pada konteks NLP yang membutuhkan pemahaman konteks linguistik yang

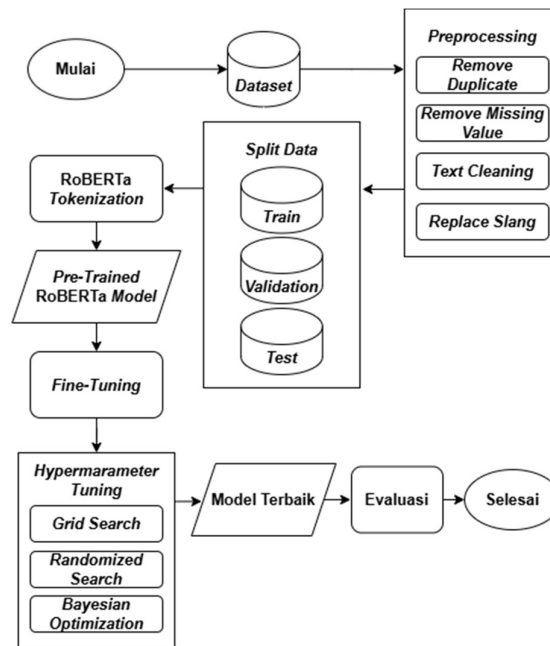
mendalam, seperti deteksi emosi [8], [9]. Meskipun RoBERTa memberikan hasil yang baik pada tugas-tugas NLP diberbagai bahasa, kinerja model ini untuk deteksi emosi masih bisa ditingkatkan melalui hyperparameter tuning [10], [11], [12]. Pemilihan *hyperparameter* yang tidak tepat dapat menyebabkan *overfitting* atau *underfitting*, dan mengurangi akurasi model dalam mendeteksi emosi secara akurat. Hal ini penting karena kesalahan dalam deteksi emosi bisa berpengaruh pada interpretasi sentimen pengguna, yang pada akhirnya mempengaruhi keputusan berbasis analisis emosi tersebut. Oleh karena itu, pendekatan sistematis untuk optimasi *hyperparameter* diperlukan untuk memaksimalkan performa RoBERTa dalam deteksi emosi.

Masalah utama yang dihadapi RoBERTa adalah ketergantungan model pada konfigurasi hyperparameter yang tepat, seperti *learning rate*, *batch size*, dan *epoch* [13]. Beberapa penelitian menunjukkan keberhasilan model-model seperti BERT dan DistilBERT dalam tugas-tugas deteksi emosi, dengan pendekatan hyperparameter tuning seperti grid search dan randomized search [14], [15], [16]. Namun, belum ada penelitian yang berfokus secara khusus pada optimasi hyperparameter untuk RoBERTa

pada tugas deteksi emosi dalam bahasa Indonesia, sehingga ini membuka peluang riset untuk menemukan metode optimasi hyperparameter yang efektif guna meningkatkan kinerja model dalam tugas ini.

Penelitian ini menerapkan metode hyperparameter tuning yang mengombinasikan pendekatan Grid Search, Randomized Search, dan Bayesian Optimization untuk optimasi RoBERTa pada deteksi emosi berbasis teks. Tujuan dari pendekatan ini adalah untuk memperoleh konfigurasi hyperparameter yang dapat meningkatkan akurasi dan efisiensi model dalam mendeteksi emosi pada teks. Hyperparameter yang akan disesuaikan mencakup learning rate, batch size, dan epoch. Dengan melakukan optimasi pada hyperparameter ini, diharapkan model dapat memberikan hasil deteksi emosi yang lebih akurat, dan untuk mendukung pengambilan keputusan yang lebih baik di berbagai bidang yang membutuhkan analisis emosi.

2. METODE



Gambar 1. Metode Penelitian

Metode penelitian yang digunakan tergambar dalam diagram yang terdapat dalam Gambar 1. Berdasarkan Gambar 1, tahapan penelitian dimulai dengan pemrosesan data yang mencakup pembersihan teks dan penanganan duplikasi. Setelah itu, data dibagi menjadi tiga bagian, yaitu data training 60%, data validation 20%, dan data testing 20%. Model RoBERTa kemudian diterapkan dengan proses tokenisasi dan *fine-tuning*. Selanjutnya, dilakukan konfigurasi hyperparameter dengan berbagai metode seperti grid search, randomized search dan bayesian optimization untuk mendapatkan model terbaik. Terakhir, model tersebut dievaluasi sebelum penelitian diselesaikan.

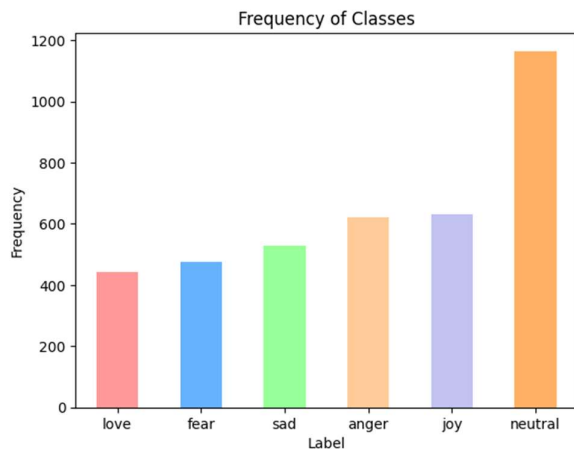
2.1. Dataset

Dataset yang digunakan dalam penelitian ini mengandung data mengenai emosi yang diekspresikan oleh masyarakat Indonesia melalui media sosial. Penelitian ini menggunakan dataset yang digunakan dalam artikel berjudul “*Emotion dataset from Indonesian public opinion*” [17]. Dataset ini terdiri atas 7080 data dan memiliki 2 atribut, yaitu “text” yang berisikan *tweet* publik pada platform X, dan “label” yang berisikan emosi dari *tweet*, seperti *anger*, *fear*, *joy*, *love*, *sad*, dan *neutral*. Contoh data teks beserta labelnya dalam deteksi emosi dapat dilihat dalam Tabel 1. Tabel 1 menampilkan beberapa contoh data yang menunjukkan bagaimana teks yang diambil dari platform media sosial dapat dikategorikan ke dalam berbagai label emosi, seperti *Sad*, *Neutral*, *Love*, *Joy*, *Anger*, dan *Fear*.

Tabel 1. Sampel *Dataset*

Text	Label
yatuhann sedih banget gue juga dritadi bacain review orang pda jelek banget habis minum bukannya langsung malah	<i>Sad</i>
semakin besar masalahnya semakin dia akan berpura bahwa dia baik sajaa	<i>Neutral</i>
baik banget baiknya juga ke semua orang mana cakep lagi wajar banyak yang suka	<i>Love</i>
selamat pagi tetap semangat tetap optimis biar bisa bigwin amin amin	<i>Joy</i>
saya benci twitter suka ke refresh sendiri padahal tadi ada <i>tweet</i> lucu jadi tidak kebaca	<i>Anger</i>
kak tertawa aku juga bingung takut akun sopi di report atau gimana huhu	<i>Fear</i>

Distribusi label emosi yang digunakan dapat dilihat dalam Gambar 2. Berdasarkan Gambar 2, terlihat bahwa label *Neutral* memiliki jumlah data terbanyak dibandingkan dengan label emosi lainnya. Hal ini menunjukkan bahwa sebagian besar teks dalam *dataset* ini cenderung tidak menunjukkan emosi yang spesifik, atau bersifat netral. Sementara itu, label emosi lainnya, seperti *Love*, *Fear*, dan *Joy*, memiliki distribusi yang lebih sedikit.



Gambar 2. Distribusi Label Dataset

2.2. Preprocessing

Tahap preprocessing dilakukan dengan *remove duplicate*, *remove missing value*, *text cleaning* dan *replace slang*. Setelah melalui tahap preprocessing, jumlah data dalam *dataset* berkurang dari 7080 menjadi 6448. *Remove duplicate* merupakan proses menghapus data duplikat. Duplikasi data bisa menyebabkan bias dalam analisis atau pelatihan model karena data yang sama berulang kali dihitung, memberikan kesan adanya pola yang sebenarnya tidak ada [18]. Proses ini dilakukan dengan mengidentifikasi baris-baris data yang identik dan menghapus salinan berlebih sehingga *dataset* menjadi lebih efisien dan akurat dalam merepresentasikan informasi yang sebenarnya. Pada tahap *remove missing value*, data yang memiliki nilai kosong atau tidak lengkap dihapus atau diisi dengan nilai tertentu [18]. Nilai yang

hilang bisa mengurangi kualitas, terutama ketika model bergantung pada adanya informasi lengkap.

Berikutnya proses *text cleaning* dilakukan untuk membersihkan data teks dari elemen-elemen yang tidak relevan [19]. Tahapan yang dilakukan meliputi penghapusan simbol-simbol khusus seperti @, #, atau tanda baca lainnya, diikuti dengan pembersihan tautan (*URL*) dan *username* yang sering muncul dalam teks media sosial. Selanjutnya, karakter emoji juga dihilangkan, bersamaan dengan penghapusan angka. Selain itu, teks dengan panjang kurang dari empat kata dihapus karena dianggap tidak memberikan informasi yang cukup.

Tabel 2. *Text Cleaning*

Sebelum	Sesudah
semoga cepat sembuh pak,,	semoga cepat sembuh pak
salam untuk keluarga besar.	salam untuk keluarga besar
tetap semangat,,	tetap semangat

Lalu terakhir *replace slang* merupakan tahap di mana bahasa gaul, singkatan, atau kata-kata informal diubah menjadi bentuk standar [20]. Hal ini bertujuan agar model dapat memahami maksud dari kata-kata yang mungkin tidak umum atau sulit dipahami. Sebagai contoh, kata seperti "btw" akan diubah menjadi "by the way" atau "gak" menjadi "tidak". Langkah ini penting dalam mempersiapkan teks agar sesuai dengan format bahasa yang dikenali oleh model bahasa.

Tabel 3. *Replace Slang*

Sebelum	Sesudah
sabar yah orang kecil kyk kita dah biasa ditinggal	sabar yah orang kecil seperti kita sudah biasa ditinggal

2.3. Split Data

Tahap *split data* dilakukan dengan membagi *dataset* menjadi tiga bagian dengan proporsi 60% untuk *training data*, 20% untuk *validation data*, dan 20% untuk *testing data* [13]. *Training data* digunakan untuk melatih model dalam mengenali pola pada data, *validation data* berfungsi untuk menyetel *hyperparameter* dan menghindari *overfitting*, serta *testing data* digunakan untuk mengevaluasi performa model akhir [21]. Distribusi data setelah dilakukan *split data* dapat dilihat pada Tabel 4.

Tabel 4. Distribusi Data Setelah *Split Data*

<i>Train</i>	3867
<i>Test</i>	1291
<i>Validation</i>	1290

2.4. RoBERTa Tokenization

Tokenisasi pada model RoBERTa menggunakan pendekatan *Byte-Pair Encoding (BPE)*, sebuah teknik tokenisasi sub-kata yang membagi teks menjadi unit-unit sub-kata alih-alih kata penuh [22]. Langkah pertama dalam tokenisasi yang dilakukan adalah pra-pemrosesan teks dengan menormalisasi teks tertentu. Tidak seperti BERT yang melakukan *lowercasing* (mengubah teks menjadi huruf kecil seluruhnya), RoBERTa mempertahankan bentuk asli huruf besar dan kecil. Selanjutnya, dalam tahap *BPE*, teks dipecah menjadi sub-kata berdasarkan frekuensi kemunculan pasangan *byte* di seluruh data, yang

memungkinkan pembentukan token dari unit yang lebih kecil. Teknik ini bermanfaat untuk menangani kata-kata baru atau langka, dengan membagi kata yang tidak dikenal menjadi sub-kata yang sudah ada dalam kamus. Setelah sub-kata terbentuk, token khusus seperti [CLS] (untuk klasifikasi) dan [SEP] (untuk membedakan segmen) ditambahkan untuk meningkatkan konteks pemahaman model.

Tabel 5. Hasil *Tokenization*

Text	Token	Input_ids
udah biasa kalah dan tersakiti	['ud', 'ah', 'a', 'Gk', 'Gt', 'ak', 'iti']	[0, 1906, 895, 9415, 102, 449, 337, 895, 20435, 326, 268, 677, 4933, 2]

2.5. Fine-Tuning

Fine-tuning merupakan metode dimana model yang sudah dilatih sebelumnya disesuaikan agar dapat menangani tugas atau *dataset* tertentu dengan lebih baik [23]. Sebelum melakukan *fine-tuning*, penting untuk mengatur nilai *hyperparameter* terlebih dahulu. Proses *fine-tuning* diawali dengan inisialisasi model RoBERTa yang telah dilatih sebelumnya. Dalam proses *fine-tuning* ini, nilai *hyperparameter* seperti *learning rate*, *batch size*, dan jumlah *epoch* diatur berdasarkan penelitian sebelumnya [13]. Setelah model RoBERTa melalui proses *fine-tuning* yang memadai, model tersebut akan memiliki pengetahuan untuk tugas khusus yang diberikan. Nilai *hyperparameter* dapat dilihat pada Tabel 6.

Tabel 6. Nilai *Hyperparameter Fine-Tuning*

Hyperparameter	Nilai
<i>Learning Rate</i>	2e-5
<i>Batch Size</i>	16
<i>Epoch</i>	10

Tabel 7. Contoh Prediksi *Fine-Tuning*

Text	Emosi Aktual	Emosi Prediksi
semangat ya aku juga sedih soalnya tapi tetap konsisten saja	<i>sad</i>	<i>sad</i>
seru banget bikin emosi	<i>joy</i>	<i>love</i>
baru saja mengirim foto manyar gresik	<i>neutral</i>	<i>neutral</i>

2.6. Hyperparameter Tuning

Hyperparameter tuning dilakukan untuk menemukan kombinasi yang optimal dan meningkatkan performa model. Untuk membandingkan model sebelum dan sesudah *hyperparameter tuning*, digunakan tiga metode utama, yaitu *grid search*, *random search*, dan *Bayesian optimization*. *Grid search* dipilih karena mudah diimplementasikan dan cocok untuk berbagai jenis model, *random search* lebih efisien dalam mencari kombinasi optimal, dan *bayesian optimization* dipilih karena lebih baik dalam memberikan hasil optimal dengan mengurangi kesalahan validasi secara efektif [13]. Rentang nilai *hyperparameter tuning* dapat dilihat pada Tabel 8.

Tabel 8. Rentang Nilai *Hyperparameter Tuning*

Hyperparameter	Nilai
<i>Learning Rate</i>	2e-5, 3e-5, 5e-5
<i>Batch Size</i>	16, 32
<i>Epoch</i>	10

Tabel 9. Contoh Prediksi Metode *Grid Search*

Text	Emosi Aktual	Emosi Prediksi
malu minta belikan paket zenius	<i>fear</i>	<i>neutral</i>
pagi udah dibikin jengkel heran	<i>anger</i>	<i>anger</i>
biasanya makan nasduk sekarang makan bubur	<i>neutral</i>	<i>neutral</i>

Tabel 10. Contoh Prediksi Metode *Random Search*

Text	Emosi Aktual	Emosi Prediksi
aku terlalu sayang kamu juga terlalu bodoh untuk memahami	<i>sad</i>	<i>sad</i>
takut patah hati gara derek	<i>fear</i>	<i>love</i>
jangan sampai rakyat benci	<i>anger</i>	<i>anger</i>

Tabel 11. Contoh Prediksi Metode *Bayesian Optimization*

Text	Emosi Aktual	Emosi Prediksi
lagi suka banget sama foto ini wish me luck	<i>love</i>	<i>love</i>
aku meriang jadi tidak nonton juga sedih	<i>sad</i>	<i>sad</i>
dapat kabar temenku positif takut jadinya	<i>fear</i>	<i>fear</i>

2.7. Evaluasi

Dalam penelitian ini, evaluasi digunakan sebagai acuan utama untuk membandingkan kinerja model setelah dilakukan *fine-tuning* dan *hyperparameter tuning*. Metode evaluasi yang dipakai mencakup *accuracy*, *precision*, *recall*, dan *F1-score* [24]. Pada penelitian ini, akurasi diperlukan untuk mengukur ketepatan sistem dalam mengklasifikasikan data secara umum, namun akurasi saja tidak cukup untuk menilai kemampuan model secara menyeluruh. Jika model selalu mengklasifikasikan kalimat sebagai negatif, akurasinya bisa tinggi, tetapi sebenarnya tidak mencerminkan kinerja yang baik. Oleh karena itu, diperlukan juga pengukuran seperti presisi, *recall*, dan *F1-score*. Presisi menunjukkan seberapa banyak prediksi positif yang benar, sedangkan *recall* mengukur sensitivitas atau kemampuan model dalam mendeteksi semua contoh yang sebenarnya positif. Untuk menggabungkan kedua matriks tersebut, *F1-score* digunakan sebagai ukuran rata-rata antara presisi dan *recall*, sehingga memberikan gambaran tentang kinerja model.

3. HASIL

3.1. Hasil Deteksi Emosi Fine-Tuning RoBERTa

Pada tahap eksperimen, dilakukan proses *fine-tuning* untuk mendeteksi emosi menggunakan konfigurasi *hyperparameter* yang terdapat pada Tabel 6. Proses *fine-tuning* dilakukan dengan nilai *learning rate* sebesar $2e-5$, *batch size* 16, dan *epoch* sebanyak 10. Setelah melalui pelatihan dengan konfigurasi tersebut, model RoBERTa mampu memberikan hasil evaluasi yang baik dalam mendeteksi emosi, dengan nilai *precision* sebesar 83.36%, *recall* 82.86%, *F1-score* 82.81%, dan *accuracy* 82.86%, seperti yang ditunjukkan pada Tabel 5. Hasil ini menunjukkan bahwa model telah dioptimalkan dengan cukup baik untuk mendeteksi emosi dalam teks berbahasa Indonesia.

Tabel 12. Nilai Evaluasi *Fine-Tuning* RoBERTa

<i>Precision</i>	82.99%
<i>Recall</i>	82.86%
<i>F1-Score</i>	82.81%
<i>Accuracy</i>	82.86%

3.2. Hyperparameter Tuning Pada Deteksi Emosi

Untuk meningkatkan performa model dalam mendeteksi emosi pada teks bahasa Indonesia, dilakukan proses *hyperparameter tuning*. Proses ini bertujuan untuk menemukan kombinasi parameter yang optimal guna memaksimalkan akurasi serta kinerja model secara keseluruhan. Tabel 8 menunjukkan rentang nilai *hyperparameter* yang diuji dalam eksperimen ini, yaitu *learning rate* dengan nilai $2e-5$, $3e-5$, dan $5e-5$, *batch size* 16 dan 32, serta jumlah *epoch* sebanyak 10. Berikutnya, akan dibahas tiga metode tuning *hyperparameter* yang digunakan, yaitu *Grid Search*, *Randomized Search*, dan *Bayesian Optimization*. Masing-masing metode ini memiliki pendekatan berbeda dalam menemukan kombinasi *hyperparameter* yang optimal bagi model.

3.2.1. Grid Search

Metode *Grid Search* digunakan untuk melakukan *hyperparameter tuning* model RoBERTa dalam mendeteksi emosi. *Grid Search* merupakan metode pencarian sistematis yang mengevaluasi semua kombinasi dari *hyperparameter* yang telah ditentukan [25]. Dalam eksperimen ini, kombinasi nilai *learning rate*, *batch size*, dan *epoch* tercantum pada Tabel 8. Dalam eksperimen ini, nilai optimal yang diperoleh untuk *learning rate* adalah $2e-5$, *batch size* 16, dan *epoch* 6, seperti pada Tabel 13. Hasil dari *Grid Search* menunjukkan peningkatan, dengan nilai *precision* sebesar 83.71%, *recall* 83.64%, *F1-score* 83.55%, dan *accuracy* 83.64% seperti yang ditampilkan pada Tabel 14.

Tabel 13. *Hyperparameter* Optimal *Grid Search*

<i>Hyperparameter</i>	Nilai
<i>Learning Rate</i>	$2e-5$
<i>Batch Size</i>	16
<i>Epoch</i>	6

Tabel 14. Nilai Evaluasi *Grid Search* RoBERTa

<i>Precision</i>	83.71%
<i>Recall</i>	83.64%
<i>F1-Score</i>	83.55%
<i>Accuracy</i>	83.64%

3.2.2. Randomized Search

Metode *Randomized Search* diterapkan untuk melakukan *hyperparameter tuning* model RoBERTa. Berbeda dengan *Grid Search* yang mengevaluasi setiap kombinasi *hyperparameter* yang mungkin, *Randomized Search* memilih secara acak beberapa kombinasi dari ruang *hyperparameter* yang telah ditentukan [26]. Pendekatan ini memungkinkan waktu komputasi yang lebih efisien, terutama ketika jumlah kombinasi *hyperparameter* yang tersedia sangat besar. Dengan melakukan *sampling* acak, *Randomized Search* memungkinkan untuk menemukan konfigurasi yang mendekati optimal tanpa memerlukan waktu dan sumber daya sebesar *Grid Search*. Dalam eksperimen ini, nilai optimal yang diperoleh untuk *learning rate* adalah $2e-5$, *batch size* 16, dan *epoch* 5, sebagaimana tercantum pada Tabel 15. Berdasarkan hasil evaluasi, model RoBERTa yang dioptimalkan menggunakan metode *Randomized Search* menunjukkan nilai *precision* sebesar 83,96%, *recall* 83,56%, *F1-score* 83,50%, dan *accuracy* 83,56%, seperti yang dapat dilihat pada Tabel 16.

Tabel 15. *Hyperparameter* Optimal *Randomized Search*

<i>Hyperparameter</i>	Nilai
<i>Learning Rate</i>	$2e-5$
<i>Batch Size</i>	16
<i>Epoch</i>	5

Tabel 16. Nilai Evaluasi *Randomized Search* RoBERTa

<i>Precision</i>	83.96%
<i>Recall</i>	83.56%
<i>F1-Score</i>	83.50%
<i>Accuracy</i>	83.56%

3.2.3. Bayesian Optimization

Metode *Bayesian Optimization* digunakan untuk melakukan *hyperparameter tuning* model RoBERTa. *Bayesian Optimization* bekerja dengan pendekatan yang berbeda dibandingkan dengan *Grid Search* dan *Randomized Search*. Metode ini memanfaatkan model probabilistik untuk memprediksi nilai fungsi objektif pada setiap iterasi dan memilih kombinasi *hyperparameter* yang dianggap memiliki peluang terbaik untuk meningkatkan performa model [27]. Dengan menggunakan pendekatan ini, *Bayesian Optimization* secara adaptif mengeksplorasi ruang *hyperparameter* berdasarkan hasil yang sudah diperoleh, sehingga memungkinkan pencarian yang lebih efisien dan efektif dalam menemukan kombinasi optimal. Dalam eksperimen ini, nilai optimal yang diperoleh untuk *learning rate* adalah $3e-5$, *batch size* 32, dan *epoch* 9, sebagaimana tercantum pada Tabel 17. Berdasarkan hasil yang diperoleh dari *tuning* dengan *Bayesian Optimization*, model RoBERTa mencapai performa dengan nilai *precision* sebesar 83.75%, *recall* 83.64%, *F1-score* 83.58%, dan *accuracy* 83.64% seperti ditampilkan pada Tabel 18.

Tabel 17. *Hyperparameter Optimal Bayesian Optimization*

<i>Hyperparameter</i>	<i>Nilai</i>
<i>Learning Rate</i>	3e-5
<i>Batch Size</i>	32
<i>Epoch</i>	9

Tabel 18. Nilai Evaluasi *Bayesian Optimization* RoBERTa

<i>Precision</i>	83.75%
<i>Recall</i>	83.64%
<i>F1-Score</i>	83.58%
<i>Accuracy</i>	83.64%

3.3. Hasil Prediksi Emosi

Tabel 19. Perbandingan Prediksi Setelah *Hyperparameter Tuning*

<i>Metode</i>	<i>Text</i>	<i>Aktual</i>	<i>Prediksi</i>
Fine-Tuning	selamat pagi luar biasa tetap semangat	joy	neutral
Grid Search	selamat pagi luar biasa tetap semangat	joy	love
Random Search	selamat pagi luar biasa tetap semangat	joy	joy
Bayesian Optimization	selamat pagi luar biasa tetap semangat	joy	joy

Pada Tabel 19, terlihat perbandingan akurasi beberapa metode setelah dilakukan *hyperparameter tuning* terhadap teks "selamat pagi luar biasa tetap semangat." Metode *Fine-Tuning* menghasilkan prediksi neutral, yang berbeda dari emosi aktual yaitu joy. Metode *Grid Search* memberikan prediksi love, yang juga tidak sesuai dengan emosi aktual. Sebaliknya, metode *Random Search* menghasilkan prediksi joy, yang sesuai dengan emosi aktual. Metode *Bayesian Optimization* juga menunjukkan prediksi yang tepat, yaitu joy.

4. PEMBAHASAN

Hasil *fine-tuning* menunjukkan model RoBERTa dapat mendeteksi emosi dengan tingkat akurasi yang cukup tinggi melalui konfigurasi *hyperparameter*, yaitu *learning rate* sebesar $2e-5$, *batch size* 16, dan *epoch* sebanyak 10. Kombinasi ini memungkinkan model untuk mempelajari pola emosi dalam data teks bahasa Indonesia dengan baik, menghasilkan nilai *precision*, *recall*, *F1-score*, dan *accuracy* yang seimbang, masing-masing di sekitar 83%. Nilai-nilai ini mengindikasikan bahwa model dapat mengenali emosi secara konsisten dengan kesalahan yang minimal. Kinerja dari *fine-tuning* masih dapat ditingkatkan lebih lanjut melalui proses *hyperparameter tuning* yang lebih mendalam, dengan tujuan untuk menemukan konfigurasi optimal yang mampu memaksimalkan performa model.

Berdasarkan Tabel 20 hingga Tabel 23, terlihat bahwa beberapa metode *hyperparameter tuning* memberikan hasil yang berbeda dalam meningkatkan kemampuan model mendeteksi enam emosi: *Sad*, *Neutral*, *Love*, *Joy*, *Anger*, dan *Fear*. Pada Tabel 20,

metode *Randomized Search* memberikan peningkatan terbesar dalam *precision* sebesar 0,97%, diikuti oleh *Bayesian Optimization* dengan peningkatan 0,76% dan *Grid Search* dengan 0,72%. Hasil ini menunjukkan bahwa *Randomized Search* lebih efektif dalam meningkatkan ketepatan model dalam mengidentifikasi emosi yang tepat.

Tabel 20. Peningkatan *Precision* dengan *Hyperparameter Tuning*

<i>Metode</i>	<i>Precision</i>	<i>Peningkatan</i>
<i>Fine-Tuning</i>	82.99%	-
<i>Grid Search</i>	83.71%	0.72%
<i>Randomized Search</i>	83.96%	0.97%
<i>Bayesian Optimization</i>	83.75%	0.76%

Tabel 21. Peningkatan *Recall* dengan *Hyperparameter Tuning*

<i>Metode</i>	<i>Recall</i>	<i>Peningkatan</i>
<i>Fine-Tuning</i>	82.86%	-
<i>Grid Search</i>	83.64%	0.78%
<i>Randomized Search</i>	83.56%	0.70%
<i>Bayesian Optimization</i>	83.64%	0.78%

Tabel 22. Peningkatan *F1-Score* dengan *Hyperparameter Tuning*

<i>Metode</i>	<i>F1-Score</i>	<i>Peningkatan</i>
<i>Fine-Tuning</i>	82.81%	-
<i>Grid Search</i>	83.55%	0.74%
<i>Randomized Search</i>	83.50%	0.69%
<i>Bayesian Optimization</i>	83.58%	0.77%

Tabel 23. Peningkatan *Accuracy* dengan *Hyperparameter Tuning*

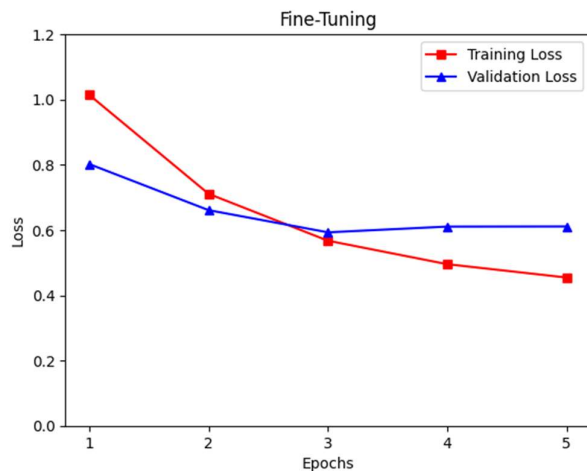
<i>Metode</i>	<i>Accuracy</i>	<i>Peningkatan</i>
<i>Fine-Tuning</i>	82.86%	-
<i>Grid Search</i>	83.64%	0.78%
<i>Randomized Search</i>	83.56%	0.70%
<i>Bayesian Optimization</i>	83.64%	0.78%

Pada Tabel 21 yang mengukur *recall*, *Randomized Search* juga menunjukkan hasil terbaik dengan peningkatan sebesar 0,70%, sementara *Grid Search* dan *Bayesian Optimization* memberikan peningkatan yang sama yaitu 0,78%. Ini menunjukkan bahwa kedua metode ini membantu model dalam mengenali lebih banyak variasi emosi secara konsisten. Kemudian, pada Tabel 22 yang memuat hasil *F1-score*, *Bayesian Optimization* mencatat peningkatan tertinggi sebesar 0,77%, diikuti oleh *Grid Search* dengan 0,74% dan *Randomized Search* sebesar 0,69%. *F1-score* yang tinggi pada *Bayesian Optimization* menunjukkan bahwa metode ini seimbang dalam meningkatkan ketepatan dan kemampuan model dalam mendeteksi emosi. Dan terakhir, pada Tabel 23 yang memperlihatkan *accuracy*, *Bayesian Optimization* dan *Grid Search* sama-sama memberikan peningkatan sebesar 0,78%, sementara *Randomized Search* sedikit lebih rendah dengan 0,60%. Secara keseluruhan, *Bayesian Optimization* adalah metode yang paling konsisten dalam memperbaiki berbagai aspek performa model, sehingga membantu model dalam mendeteksi emosi seperti *Sad*, *Neutral*, *Love*, *Joy*, *Anger*, dan *Fear* dengan lebih akurat.

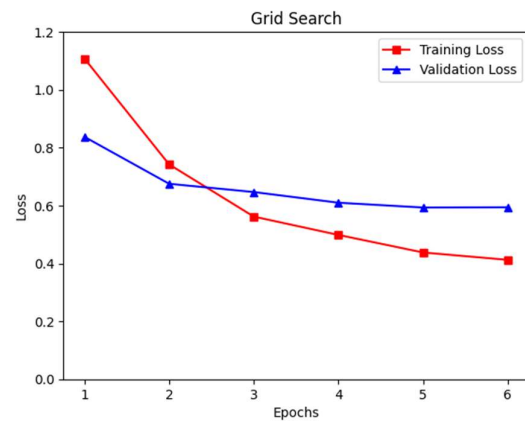
Berdasarkan Gambar 3 hingga Gambar 6, terlihat bahwa *training loss* menurun secara signifikan pada semua metode *hyperparameter tuning* (*fine-tuning*, *grid search*, *randomized search*, dan *bayesian optimization*). Namun, *validation loss*

cenderung sulit turun secara signifikan, terutama pada *randomized search*, yang disebabkan oleh kualitas dataset yang kurang optimal. Pada Gambar 6, *bayesian optimization* menunjukkan kesesuaian yang lebih baik antara *training loss* dan *validation loss*, tetapi kualitas dataset yang lebih baik diperlukan untuk meningkatkan generalisasi model pada data validasi.

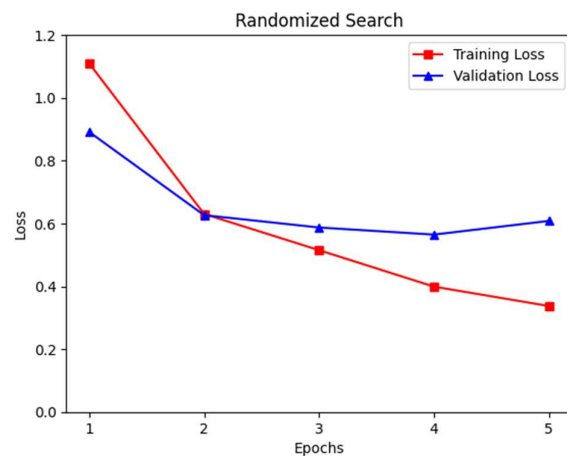
Dalam penelitian ini, model RoBERTa dikembangkan untuk mendeteksi enam label emosi, yaitu *anger*, *fear*, *joy*, *love*, *sad*, dan *neutral*. Pendekatan ini berbeda dari penelitian-penelitian sebelumnya yang umumnya hanya menggunakan dua sampai empat label emosi, seperti positif-negatif atau senang-sedih-marah-takut [28], [29], [30]. Dengan menambah jumlah label emosi, penelitian ini memberikan analisis yang lebih mendalam terhadap emosi pada teks Bahasa Indonesia. Penggunaan enam label emosi ini memberikan tantangan tambahan, karena model harus mampu membedakan lebih banyak variasi emosi yang saling berdekatan. Hasil yang diperoleh menunjukkan bahwa dengan konfigurasi hyperparameter tuning yang optimal, model RoBERTa mampu mengidentifikasi berbagai emosi dengan akurasi yang cukup baik. Oleh karena itu, penelitian ini memberikan kontribusi tidak hanya dalam pengembangan deteksi emosi dengan variasi label emosi yang lebih beragam, tetapi juga dalam meningkatkan performa model melalui *hyperparameter tuning*. Penerapan metode *hyperparameter tuning* seperti *Grid Search*, *Random ized Search*, dan *Bayesian Optimization* telah terbukti efektif dalam meningkatkan nilai *precision*, *recall*, *F1-score*, dan *accuracy*, sehingga menunjukkan pentingnya pengaturan *hyperparameter* yang tepat untuk mencapai hasil klasifikasi emosi yang lebih akurat pada teks Bahasa Indonesia.



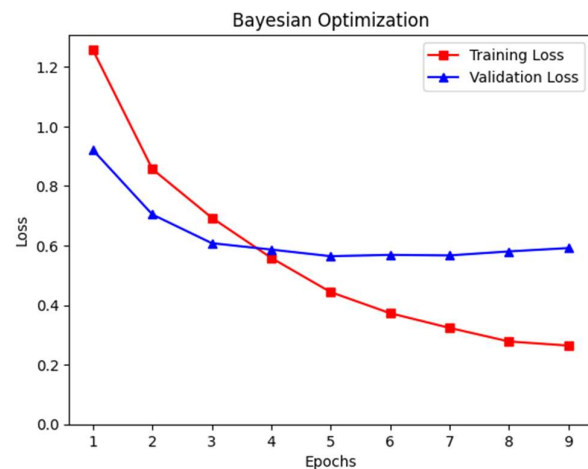
Gambar 3. Training loss dan validation loss fine-tuning



Gambar 4. Training loss dan validation loss grid search



Gambar 5. Training loss dan validation loss randomized search



Gambar 6. Training loss dan validation loss bayesian optimization

5. KESIMPULAN

Metode *hyperparameter tuning* dengan *Bayesian Optimization* menunjukkan kinerja yang lebih baik di antara dua metode lainnya. Metode ini mencapai nilai *recall*, *f1-score*, dan *accuracy* tertinggi dalam mendeteksi emosi berbasis teks, dengan nilai

hyperparameter optimal 3×10^{-5} untuk *learning rate*, 32 untuk *batch size*, dan 9 untuk jumlah *epoch*. Jika dilihat dari hasil keseluruhan eksperimen *hyperparameter tuning*, setiap metode yaitu *Grid Search*, *Randomized Search*, dan *Bayesian Optimization* menunjukkan peningkatan performa model RoBERTa dalam mendeteksi emosi, dibandingkan dengan hanya menerapkan fine-tuning. Setiap metode *hyperparameter tuning* memiliki keunggulan masing-masing dalam meningkatkan performa model RoBERTa untuk mendeteksi emosi. *Randomized Search* memberikan peningkatan terbaik pada *precision*. *Grid Search*, di sisi lain, memberikan hasil yang stabil di semua metrik karena menguji setiap kombinasi *hyperparameter*. Sementara itu, *Bayesian Optimization* memberikan hasil yang paling baik, dengan memprediksi kombinasi *hyperparameter* terbaik berdasarkan hasil evaluasi sebelumnya sehingga membuat iterasi lebih sedikit.

Meskipun dataset yang digunakan saat ini sudah mampu menghasilkan model dengan performa yang cukup baik, ketidakseimbangan pada distribusi kelas dapat menyebabkan bias yang berdampak pada akurasi prediksi, terutama pada kelas minoritas. Untuk penelitian berikutnya, sebaiknya mempertimbangkan penggunaan metode *balancing* untuk memastikan distribusi kelas yang lebih merata. Salah satu metode yang bisa dipertimbangkan adalah *oversampling* pada kelas minoritas, misalnya dengan *SMOTE* (Synthetic Minority Over-sampling Technique) atau *ADASYN* (Adaptive Synthetic Sampling), serta *undersampling* pada kelas mayoritas.

DAFTAR PUSTAKA

- [1] W. Anggriyani and M. Fakhriya, "Analisis Sentimen Program Makan Gratis Pada Media Sosial X Menggunakan Metode NLP," *Journal of Computer System and Informatics (JoSYC)*, vol. 5, no. 4, pp. 1033–1042, 2024, doi: [10.47065/josyc.v5i4.5826](https://doi.org/10.47065/josyc.v5i4.5826).
- [2] T. I. Z. M. Putra, S. Suprpto, and A. F. Bukhori, "Model Klasifikasi Berbasis Multiclass Classification dengan Kombinasi Indobert Embedding dan Long Short-Term Memory untuk Tweet Berbahasa Indonesia," *Jurnal Ilmu Siber dan Teknologi Digital*, vol. 1, no. 1, pp. 1–28, Nov. 2022, doi: [10.35912/jisted.v1i1.1509](https://doi.org/10.35912/jisted.v1i1.1509).
- [3] A. Hamzah and R. Yanwastika Ariyana, "Klasifikasi Emosi Berbasis Emolex dari Komentar Evaluasi Akademik Mahasiswa Emolex-Based Classification of Emotions from Academic Evaluation Comments," *Techno.COM*, vol. 23, no. 2, pp. 457–468, May 2024, doi: [10.62411/tc.v23i2.10058](https://doi.org/10.62411/tc.v23i2.10058).
- [4] S. Imron, E. I. Setiawan, and J. Santoso, "Deteksi Aspek Review E-Commerce Menggunakan IndoBERT Embedding dan CNN," *Journal of Intelligent System and Computation*, vol. 5, no. 1, pp. 10–16, Apr. 2023, doi: [10.52985/insyst.v5i1.267](https://doi.org/10.52985/insyst.v5i1.267).
- [5] E. Y. Hidayat and D. Handayani, "Penerapan 1D-CNN untuk Analisis Sentimen Ulasan Produk Kosmetik Berdasar Female Daily Review," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 8, no. 3, pp. 153–163, Jan. 2023, doi: [10.25077/teknosi.v8i3.2022.153-163](https://doi.org/10.25077/teknosi.v8i3.2022.153-163).
- [6] E. Y. Hidayat, R. W. Hardiansyah, and A. Affandy, "Analisis Sentimen Twitter untuk Menilai Opini Terhadap Perusahaan Publik Menggunakan Algoritma Deep Neural Network," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 7, no. 2, pp. 108–118, Sep. 2021, doi: [10.25077/teknosi.v7i2.2021.108-118](https://doi.org/10.25077/teknosi.v7i2.2021.108-118).
- [7] A. Ansori, "Kepribadian dan Emosi," *Jurnal Literasi Pendidikan Nusantara*, vol. 1, no. 1, pp. 41–54, Jun. 2020.
- [8] Y. K. Wiciaputra, J. C. Young, and A. Rusli, "Bilingual text classification in english and indonesian via transfer learning using XLM-RoBERTa," *International Journal of Advances in Soft Computing and its Applications*, vol. 13, no. 3, pp. 72–87, 2021, doi: [10.15849/ijasca.211128.06](https://doi.org/10.15849/ijasca.211128.06).
- [9] B. Richardson and A. Wicaksana, "Comparison Of Indobert-Lite And Roberta In Text Mining For Indonesian Language Question Answering Application," *International Journal of Innovative Computing, Information and Control*, vol. 18, no. 6, pp. 1719–1734, Dec. 2022, doi: [10.24507/ijicic.18.06.1719](https://doi.org/10.24507/ijicic.18.06.1719).
- [10] M. Evtimova, "Hyperparameter Tuning for Address Validation using Optuna," *WSEAS. Transactions on Computer Research*, vol. 12, pp. 105–111, Nov. 2023, doi: [10.37394/232018.2024.12.10](https://doi.org/10.37394/232018.2024.12.10).
- [11] R. Khusuma, W. Maharani, and P. H. Gani, "Personality Detection On Twitter User With RoBERTa," *Jurnal Media Informatika Budidarma*, vol. 7, no. 1, p. 542, Feb. 2023, doi: [10.30865/mib.v7i1.5598](https://doi.org/10.30865/mib.v7i1.5598).
- [12] S. Akrah, "DuluthNLP at SemEval-2021 Task 7: Fine-Tuning RoBERTa Model for Humor Detection and Offense Rating," in *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, Aug. 2021, pp. 1196–1203.
- [13] A. C. N. Simanjuntak et al., "Studi dan Analisis Hyperparameter Tuning IndoBERT Dalam Pendeteksian Berita Palsu," *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, vol. 13, no. 1, pp. 60–67, 2024, doi: [10.22146/jnteti.v13i1.8532](https://doi.org/10.22146/jnteti.v13i1.8532).
- [14] D. A. Anggoro and S. S. Mukti, "Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 6, pp. 198–207, Dec. 2021, doi: [10.22266/ijies2021.1231.19](https://doi.org/10.22266/ijies2021.1231.19).
- [15] I. S. Kervanci, M. F. Akay, and E. Özceylan, "Bitcoin Price Prediction Using Lstm, Gru And Hybrid Lstm-Gru With Bayesian Optimization, Random Search, And Grid Search For The Next Days," *Journal of Industrial and Management Optimization*, vol. 20, no. 2, pp. 570–588, Feb. 2024, doi: [10.3934/jimo.2023091](https://doi.org/10.3934/jimo.2023091).
- [16] S. Prabu, B. Thiyaneswaran, M. Sujatha, C. Nalini, and S. Rajkumar, "Grid Search for Predicting Coronary Heart Disease by Tuning Hyper-Parameters," *Computer Systems Science and Engineering*, vol. 43, no. 2, pp. 737–749, 2022, doi: [10.32604/csse.2022.022739](https://doi.org/10.32604/csse.2022.022739).
- [17] Riccosan, K. E. Saputra, G. D. Pratama, and A. Chowanda, "Emotion dataset from Indonesian public opinion," *Data Brief*, vol. 43, no. 108465, pp. 1–6, Jul. 2022.
- [18] V. Çetin and O. Yıldız, "A comprehensive review on data preprocessing techniques in data analysis,"

- Pamukkale University Journal of Engineering Sciences*, vol. 28, no. 2, pp. 299–312, 2022, doi: [10.5505/pajes.2021.62687](https://doi.org/10.5505/pajes.2021.62687).
- [19] K. Rahayu, V. Fitria, D. Septhya, R. Rahmaddeni, and L. Efrizoni, “Klasifikasi Teks untuk Mendeteksi Depresi dan Kecemasan pada Pengguna Twitter Berbasis Machine Learning: Text Classification for Detecting Depression and Anxiety among Twitter Users based on Machine Learning,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, pp. 108–114, 2023.
- [20] M. Ayu and A. Muhendra, “Preprocessing of Slang Words for Sentiment Analysis on Public Perceptions in Twitter,” 2023. doi: [10.5772/intechopen.113725](https://doi.org/10.5772/intechopen.113725).
- [21] H. Imaduddin, F. Y. A’la, and Y. S. Nugroho, “Sentiment Analysis in Indonesian Healthcare Applications using IndoBERT Approach,” *IJACSA*, vol. 14, no. 8, pp. 113–117, 2023.
- [22] K. L. Tan, C. P. Lee, K. S. M. Anbananthen, and K. M. Lim, “RoBERTa-LSTM: A Hybrid Model for Sentiment Analysis With Transformer and Recurrent Neural Network,” *IEEE Access*, vol. 10, pp. 21517–21525, 2022, doi: [10.1109/ACCESS.2022.3152828](https://doi.org/10.1109/ACCESS.2022.3152828).
- [23] N. Gupta, “A Pre-Trained Vs Fine-Tuning Methodology in Transfer Learning,” *Journal of Physics*, vol. 1947, no. 1, pp. 1–8, 2021.
- [24] H. Wang, L. Zhang, K. Yin, H. Luo, and J. Li, “Landslide identification using machine learning,” *Geoscience Frontiers*, vol. 12, no. 1, pp. 351–364, 2021, doi: [10.1016/j.gsf.2020.02.012](https://doi.org/10.1016/j.gsf.2020.02.012).
- [25] S. F. M. Radzi, M. K. A. Karim, M. I. Saripan, M. A. A. Rahman, I. N. C. Isa, and M. J. Ibahim, “Hyperparameter tuning and pipeline optimization via grid search method and tree-based autoML in breast cancer prediction,” *J Pers Med*, vol. 11, no. 10, Oct. 2021, doi: [10.3390/jpm11100978](https://doi.org/10.3390/jpm11100978).
- [26] A. R. M. Rom, N. Jamil, and S. Ibrahim, “Multi objective hyperparameter tuning via random search on deep learning models,” *Telkommika*, vol. 22, no. 4, pp. 956–968, Aug. 2024.
- [27] H. Cho, Y. Kim, E. Lee, D. Choi, Y. Lee, and W. Rhee, “Basic Enhancement Strategies When Using Bayesian Optimization for Hyperparameter Tuning of Deep Neural Networks,” *IEEE Access*, vol. 8, pp. 52588–52608, 2020, doi: [10.1109/ACCESS.2020.2981072](https://doi.org/10.1109/ACCESS.2020.2981072).
- [28] U. Khairani, V. Mutiawani, and H. Ahmadian, “Pengaruh Tahapan Preprocessing Terhadap Model Indobert Dan Indobertweet Untuk Mendeteksi Emosi Pada Komentar Akun Berita Instagram,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 4, pp. 887–894, Aug. 2024, doi: [10.25126/jtiik.1148315](https://doi.org/10.25126/jtiik.1148315).
- [29] S. Dharmawan,) Viny, C. Mawardi,) Novario, and J. Perdana, “Klasifikasi Ujaran Kebencian Menggunakan Metode FeedForward Neural Network (IndoBERT),” pp. 1–6, 2023.
- [30] Wildan Amru Hidayat and V. R. S. Nastiti, “Perbandingan Kinerja Pre-Trained Indobert-Base Dan Indobert-Lite Pada Klasifikasi Sentimen Ulasan Tiktok Tokopedia Seller Center Dengan Model Indobert,” *Jurnal Sistem Informasi*, vol. 11, no. 2, pp. 13–20, Sep. 2024, doi: [10.30656/jsii.v11i2.9168](https://doi.org/10.30656/jsii.v11i2.9168).