TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

Analisis Komparatif Konfigurasi Multilayer Perceptron pada Classifier Head RoBERTa untuk Klasifikasi Ujaran Kebencian

Ikhwan Habibi, Surya Agustian*, Jasril, Muhammad Affandes

Fakultas Sains dan Teknologi, Program Studi Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia
Email: [1]11950111692@students.uin-suska.ac.id, [2,*]surya.agustian@uin-suska.ac.id, [3]jasril@uin-suska.ac.id, [4]affandes@uin-suska.ac.id
Email Penulis Korespondensi: surya.agustian@uin-suska.ac.id

Abstrak–Penyebaran *hate speech* dan *offensive language* di media sosial meningkatkan kebutuhan akan sistem klasifikasi otomatis yang akurat. Meskipun RoBERTaForSequenceClassification telah banyak digunakan untuk berbagai tugas klasifikasi teks, pengaruh konfigurasi internal *classifier head* bawaannya terhadap performa klasifikasi masih belum banyak dievaluasi secara sistematis. Sebagai Sebagai kontribusi utama, penelitian ini mengevaluasi secara terkontrol 32 konfigurasi classifier head berbasis multilayer perceptron, dengan variasi jumlah hidden layer, fungsi aktivasi, dan dropout, dibandingkan classifier head bawaan, pada dataset HASOC 2021 bahasa Inggris untuk dua subtask: klasifikasi biner dan klasifikasi multikelas.. *Setiap konfigurasi dievaluasi* menggunakan *Stratified* 5-*Fold Cross-Validation* dengan Macro-F1, *kemudian konfigurasi dengan performa validasi tertinggi diuji ulang pada data uji independen.*. Pada klasifikasi biner, konfigurasi terbaik memperoleh Macro-F1 uji sebesar 80.90%, sekitar 0.3 poin persentase lebih tinggi dibandingkan *baseline* sebesar 80.59%. Pada tugas klasifikasi multikelas, konfigurasi dengan performa validasi tertinggi justru memperoleh Macro-F1 uji sebesar 65.68%, sekitar 0,4 poin persentase lebih rendah dibandingkan *baseline* sebesar 66.11%, menunjukkan bahwa keunggulan yang tampak selama validasi silang tidak selalu bertahan pada data uji. Analisis lebih lanjut menemukan bahwa penambahan *hidden layer* yang berlebihan disertai kompresi dimensi yang agresif dapat menurunkan performa secara tajam pada tugas multikelas. Temuan ini menunjukkan bahwa pengaruh konfigurasi *classifier head* bersifat kecil dan bergantung pada karakteristik tugas, sehingga evaluasi sistematis tetap diperlukan sebelum suatu konfigurasi diadopsi sebagai pengganti *classifier head* bawaan dalam proses *fine-tuning* model berbasis RoBERTa**.**

Kata Kunci: Ujaran Kebencian; Bahasa Ofensif; RoBERTa; *Classifier Head*; *Multilayer Perceptron*

Abstract–The widespread dissemination of hate speech and offensive language on social media has increased the demand for accurate automated text classification systems. Although RoBERTaForSequenceClassification has been widely used for various for text classification task, the effect of its default classifier head configuration on classification performance has not yet been systematically evaluated. As the main contribution, this study conducts a controlled evaluation of 32 multilayer perceptron (MLP)-based classifier head configurations, varying the number of hidden layers, activation functions, and dropout rates, against the default classifier head on the English HASOC 2021 dataset for two subtasks: binary and multiclass classification. Each configuration was evaluated using Stratified 5-Fold Cross-Validation with Macro-F1 as the evaluation metric, after which the best-performing configuration was further evaluated on an independent test set. For the binary task, the best configuration achieved a test Macro-F1 of 80.90%, about 0.3 percentage points higher than the baseline's 80.59%. For the multiclass task, the configuration with the highest validation performance instead achieved a test Macro-F1 of 65.68%, about 0.4 percentage points lower than the baseline's 66.11%, showing that an advantage observed during cross-validation does not always hold on the test set. Further analysis revealed that excessively deep hidden layers combined with aggressive dimensional compression can sharply degrade performance on the multiclass task. These findings indicate that the effect of classifier head configuration is small and task-dependent, so systematic evaluation remains necessary before adopting a given configuration in place of the default classifier head when fine-tuning RoBERTa-based models.

Keywords: Hate Speech; Offensive Language; Roberta; Classifier Head; Multilayer Perceptron

1. PENDAHULUAN

Penyebaran ujaran kebencian (*hate speech*) dan bahasa ofensif (*offensive language*) di berbagai platform media sosial terus menjadi permasalahan yang belum terselesaikan dalam ruang digital. Konten semacam ini tidak hanya menggangu kualitas interaksi antar pengguna, tetapi juga dapat memperburuk polarisasi sosial dan menghambat terciptanya lingkungan daring yang aman dan inklusif (Cinelli et al., 2021; Subramanian et al., 2023). Volume konten yang diproduksi pengguna terus bertumbuh sehingga moderasi secara manual tidak lagi dapat dilakukan secara konsisten dalam skala besar, dan kebutuhan terhadap sistem deteksi otomatis yang andal menjadi semakin mendesak (Zaghouani et al., 2024).

Membangun sistem deteksi yang andal menghadapi sejumlan tantangan yang tidak trival. Teks media sosial cenderung singkat, informal, dan memiliki variasi penulisan yang tinggi (Yin & Zubiaga, 2021). Ujaran bermasalah juga tidak selalu disampaikan secara eksplisit. Sindiran, metafora, sarkasme, dan bahasa berkode membuat maknanya sulit dikenali hanya dari kata-kata yang muncul secara literal (ElSherief et al., 2021; N. Ocampo et al., 2023). Interpretasinya sering bergantung pada konteks percakapan, target ujaran, dan relasi makna antarbagian teks (Pérez et al., 2023). Dari sisi pengembangan model, tantangan bertambah karena keterbatasan data beranotasi, inkonsistensi kualitas pelabelan, dan distribusi kelas yang tidak seimbang pada dataset yang (Mnassri et al., 2024; Roychowdhury & Gupta, 2023).

Berbagai pendekatan telah dikembangkan untuk mengatasi tantangan tersebut, mulai dari metode berbasis fitur manual seperti TF-IDF dan *n-gram* (Subramanian et al., 2023), model *deep learning*, seperti CNN, RNN, LSTM, dan

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

BiLSTM, yang menawarkan penodelan representasi yang lebih fleksibel tanpa bergantung pada rekayasa fitur manual (Subramanian et al., 2023; Yadav et al., 2024), hingga model berbasis Transformers yang memanfaatkan mekanisme *self-attention* untuk menangkap depedensi kontekstual secara lebih efektif (Vaswani et al., 2017). Diantara model-model tersebut, RoBERTa dikembangkan melalui optimasi prosedur pelatihan BERT dan telah banyak diadopsi sebagai *backbone* berperforma kompetitif pada berbagai tugas klasifikasi teks termasuk deteksi *hate speech* (Modha et al., 2021).

RoBERTa umumnya diimplementasikan untuk tugask klasifikasi teks melalui RoBERTaForSequenceClassification, yang menggabungkan RoBERTaModel sebagai *encoder* dengan *classifier head* bawaan berbasis *multilayer perceptron* (MLP). *Classifier head* ini tersusun atas lapisan *dropout*, lapisan *linear* berdimensi 768, fungsi aktivasi Tanh, lapisan *dropout*, dan lapisan keluaran (Liu et al., 2019). Konfigurasi ini bersifat generik, disediakan untuk beragam tugas *sequence classification* secara umum, bukan dirancang khusus untuk karakteristik tugas tertentu. Kondisi ini menyisakan pertanyaan terbuka bagi klasifikasi *hate speech*. Sifat implisit ujaran bermasalah, sebagaimana telah diuraikan sebelumnya, dapat diduga membutuhkan transformasi representasi yang cukup kaya pada tahap klasifikasi, sehingga *classifier head* dengan struktur *hidden layer* tunggal yang dangkal berpotensi menjadi titik pembatas. Namun sebaliknya, ketika *encoder* RoBERTa turut dilatih penuh melalui proses *fine-tuning*, representasi kontekstual pada lapisan akhir *encoder* sudah relatif kaya. Penambahan kompleksitas pada *classifier head* dalam kondisi ini justru menambah jumlah parameter yang harus dipelajari dari kondisi acak dengan data terbatas, sehingga berisiko mengganggu stabilitas dan kemampuan generalisasi model. Kedua kemungkinan tersebut berlawanan arah, dan efektivitasnya diperkirakan turut ditentukan oleh pilihan fungsi aktivasi maupun nilai *dropout* yang menyertai struktur tersebut, mengingat kedua komponen ini dikenal luas berpengaruh terhadap kemampuan representasi dan generalisasi jaringan (Dubey et al., 2022; Salehin & Kang, 2023).

Penelitian pada HASOC 2021 menunjukkan bahwa peningkatan performa klasifikasi dapat dilakukan melalui strategi yang beragam. Bölücü & Canbay (2021) menggunakan Graph Convolutional Network (GCN) dan memperoleh Macro-F1 sebesar 0.8215 pada English Subtask A dan menjadi salah satu sistem dengan performa terbaik pada kompetisi tersebut. Pendekatan berbasis Transformer digunakan oleh Glazkova et al. (2021) melalui proses *fine-tuning* pada beberapa model pra-latih, termasuk Twitter-RoBERTa, penelitian tersebut menerapkan pendekatan *one-vs-rest* untuk membedakan kelas, dengan hasil Macro-F1 sebesar 0.8199 pada English Subtask A dan 0.6577 pada Subtask B. Mitra & Sankhala (2021) berfokus pada penanganan ketidakseimbangan kelas melalui memodifikasi *cross-entropy loss* yang dikombinasikan dengan beberapa model bahasa pra-latih, penelitian tersebut menujukan bahwa BERT-*large cased* memberikan kinerja terbaik dengan memperoleh Macro-F1 sebesar 0.8089. Penelitian lain dilakukan oleh Kui (2021) dengan mengevaluasi beberapa model Transformer, seperti BERT, ALBERT, mBERT, DeBERTa, XLNet, dan SqueezeBERT, kemudian mengombinasikan representasi yang dihasilkan dengan arsitektur RNN, BiLSTM, dan TextCNN sebelum proses klasifikasi dilakukan, memperoleh Macro-F1 sebesar 0.8030 pada model kombinasi DeBERTa + BiLSTM. Agustian et al. (2021) menerapkan pendekatan transfer learning berbasis BERT dan FastText, hasil yang diperoleh mencapai Macro-F1 sebesar 0.8024 pada model BERT + NN. Berbagai penelitian tersebut meningkatkan performa melalui modifikasi pada komponen model yang berbeda, seperti pemilihan *backbone*, fungsi *loss*, maupun arsitektur klasifikasi. Namun,belum ada yang menjadikan konfigurasi internal MLP pada *classifier head* bawaan RoBERTa sebagai objek kajian utama.

Di luar konteks HASOC 2021, beberapa penelitian memberikan indikasi bahwa konfigurasi *classifier head* bukan keputusan yang bisa diabaikan begitu saja. Masud et al. (2024) menemukan bahwa perbedaan kompleksitas *classifier head* dapat menghasilkan perbedaan performa pada deteksi *hate speech*, meskipun evaluasi tersebut dilakukan dengan backbone yang dibekukan sehingga kondisinya berbeda dari skenario *fine-tuning* penuh yang umum digunakan. Kenneweg et al. (2022) di sisi lain memasukkan jumlah *dense layer* dan jumlah *neuron* ke dalam ruang pencarian arsitektur *classifier head* berbasis BERT, mengindikasikan bahwa komponen-komponen tersebut merupakan keputusan perancangan yang dapat mempengaruhi hasil klasifikasi, meski dilakukan lewat pendekatan pencarian arsitektur, bukan evaluasi terkontrol. Karena bukti dari kedua penelitian tersebut masih bersifat tidak langsung terhadap konteks penelitian ini, sejauh mana variasi struktur *hidden layer*, fungsi aktivasi, dan nilai *dropout* pada *classifier head* bawaan RoBERTa memengaruhi performa klasifikasi *hate speech* dalam kondisi *fine-tuning* penuh belum pernah dievaluasi secara terkontrol. Kesenjangan inilah yang menjadi dasar penelitian ini..

Berangkat dari uraian tersebut, Penelitian ini bertujuan untuk mengkaji apakah modifikasi struktur *hidden layer* pada *classifier head*, yang dalam penelitian ini merepresentasikan kombinasi kedalaman jaringan dan pola penyempitan (kompresi) jumlah *neuron* secara bertahap, memengaruhi performa klasifikasi secara konsisten pada tugas klasifikasi. Penelitian ini juga mengkaji sejauh mana pengaruh fungsi aktivasi dan nilai *dropout* bergantung pada struktur *hidden layer* yang digunakan. Untuk menjawab hal tersebut, penelitian ini membandingkan performa *baseline* RoBERTaForSequenceClassification dengan 32 konfigurasi *classifier head* berbasis MLP yang memvariasikan struktur *hidden* layer, fungsi aktivasi, dan nilai *dropout* secara sistematis, menggunakan dataset HASOC 2021 bahasa inggris. Dataset ini dipilih karena menyediakan dua *subtask* dengan karakteristik yang berbeda dalam satu *benchmark* yang sama, sehingga memungkinkan evaluasi dilakukan pada dua jenis tugas sekaligus dalam kondisi yang sebanding. Seluruh konfigurasi dievaluasi menggunakan *Stratified* 5-*Fold Cross-Validation* dengan Macro-F1 sebagai metrik utama. Seluruh konfigurasi dilatih dalam pengaturan yang seragam agar perbedaan hasil dapat dikaitkan dengan perbedaan konfigurasi *classifier head* yang diuji, bukan dengan faktor pelatihan lainnya. Kontribusi penelitian ini terletak pada evaluasi terkontrol terhadap konfigurasi classifier head RoBERTa dalam kondisi *fine-tuning* penuh,

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

sebuah kondisi yang berbeda dari penelitian classifier head sebelumnya yang menggunakan backbone dibekukan atau pendekatan pencarian arsitektur.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilakukan melalui lima tahapan utama yang disusun secara sistematis untuk memastikan proses eksperimen berjalan secara terkontrol dan hasilnya dapat diinterpretasikan secara terarah. Tahapan tersebut meliputi pengumpulan data, praproses teks, pemodelan dan rancangan eksperimen, pelatihan model, dan evaluasi model. Alur tahapan penelitian ditunjukkan pada Gambar 1.

Gambar 1. Alur tahapan penelitian

Berdasarkan Gambar 1, tahapan pertama adalah pengumpulan dataset HASOC 2021 bahasa Inggris. Tahap kedua adalah praproses teks untuk membersihkan dan menyeragamkan teks mentah. Tahap ketiga adalah pemodelan dan menetapkan konfigurasi yang akan diuji, dan menyusun prosedur evaluasi. Tahap keempat adalah melatih seluruh konfigurasi dalam pengaturan yang seragam menggunakan stratified 5-*fold cross-validation*. Tahap kelima adalah evaluasi performa setiap konfigurasi menggunakan Macro-F1.

2.2 Pengumpulan Data

Penelitian ini menggunakan dataset HASOC 2021 bahasa Inggris dari *shared tas*k *Hate Speech and Offensive Content Identification* pada FIRE 2021 (Modha et al., 2021). Dataset ini disusun melalui *targeted sampling*, dan setiap *tweet* dianotasi oleh minimal dua anotator. Penelitian ini mecakup dua *subtask*. Subtask A merupakan klasifikasi biner antara dua kelas, yaitu HOF dan NOT. Dengan HOF merepresentasikan *tweet* yang mengandung *hate speech*, *offensive language*, atau *profane content* dan NOT *tweet* yang tidak mengandung konten tersebut. Subtask B merupakan klasifikasi multikelas dengan empat kelas, *HATE* untuk *tweet* yang mengandung ujaran kebencian, OFFN *tweet* yang mengandung bahasa ofensif, dan PRFN *tweet* yang mengandung konten profan, dan NONE untuk *tweet* yang tidak termasuk ke dalam kategori tersebut. Distribusi data yang digunakan ditampilkankan pada Tabel 1.

Tabel 1. Distribusi data HASOC 2021

Subtask A	total	HOF		NOT	
Train	3843	2501		1342	
Test	1281	798		483	
Subtask B	total	HATE	NONE	OFFN	PRFN
Train	3843	683	1342	622	1196
Test	1281	224	483	195	379

Tabel 1, menunjukkan ketimpangan distribusi kelas pada kedua *subtask*. Penelitian ini tidak menerapkan teknik penanganan ketidakseimbangan kelas seperti *oversampling*, *undersampling*, atau pembobotan kelas pada fungsi loss. Kompensasi ketidakseimbangan ini diserahkan sepenuhnya pada pemilihan Macro-F1 sebagai metrik evaluasi, yang menimbang setiap kelas secara setara tanpa memandang jumlah sampelnya. Performa antar konfigurasi *classifier head* dengan demikian tetap dapat dibandingkan pada kondisi data yang sama, tanpa tambahan variabel dari mekanisme resampling atau pembobotan.

2.3 Praproses Teks

Praproses teks dilakukan untuk membersihkan dan menyeragamkan teks mentah sebelum digunakan sebagai masukan model. Teks Twitter mentah mengandung sejumlah elemen yang tidak relevan secara semantik dan berpotensi menambah noise pada representasi yang dihasilkan model. Tahapannya meliputi penghapusan URL, normalisasi nama pengguna menjadi @user, penghapusan simbol pagar (#) pada *hastag*, dan penghapusan spasi berlebih. Teks yang telah melalui tahap praproses kemudian ditokenisasi menggunakan tokenizer RoBERTa-*base* dengan panjanga maksimum 128 token untuk menghasilkan format masukan model berupa *input_ids* dan *attention_mask*.

2.4 Arsitektur Model

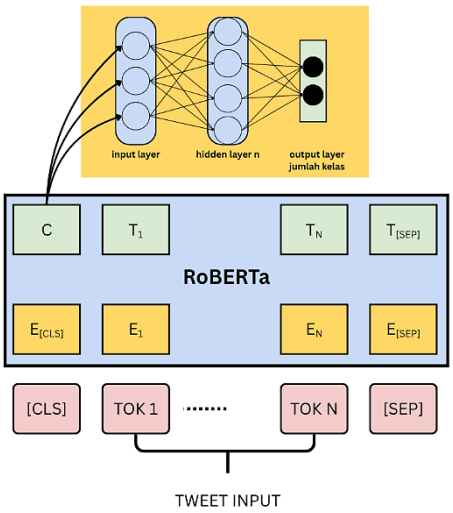
Penelitian ini menggunakan RoBERTaForSequenceClassification sebagai model dasar untuk tugas klasifikasi. Model tersebut terdiri atas RoBERTa-*base* sebagai *encoder* dan sebuah *classifier head* untuk menghasilkan prediksi kelas. Tweet yang telah ditokenisasi diproses oleh RoBERTa-*base* untuk menghasilkan representasi kontekstual bagi setiap

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

token. Representasi token khusus <s> atau CLS digunakan sebagai representasi keseluruhan tweet dan diteruskan ke *classifier head* untuk menghasilkan logit kelas (Liu et al., 2019).

$$H = \text{"}RoBERTa\text{"}(X) \qquad h = H_0 \tag{1}$$

Pada Persamaan 1, (H) merupakan keluaran kontekstual RoBERTa dan (H_0) menunjukkan representasi token <s> dengan memiliki dimensi 768. Alur model dari masukan teks hingga keluaran kelas ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur klasifikasi RoBERTa dan *classifier head*

Pada model baseline, *classifier head* bawaan RoBERTaForSequenceClassification digunakan tanpa modifikasi. Pada konfigurasi eksperimen, hanya *classifier head* yang diganti dengan beberapa konfigurasi MLP, sedangkan *encoder* RoBERTa-*base* serta prosedur pelatihan dipertahankan tetap. Dengan demikian, perbedaan performa yang diamati lebih dapat diatribusikan pada variasi konfigurasi *classifier head*. Rincian konfigurasi yang diuji disajikan pada subbab rancangan eksperimen.

2.5 Rancangan Eksperimen

Penelitian ini menguji 32 konfigurasi MLP terhadap *baseline*, dibentuk dari tiga variasi: struktur *hidden layer*, fungsi aktivasi, dan nilai *dropout*. Struktur *hidden layer* yang diuji terdiri atas empat pilihan yang dipilih secara purposif untuk mewakili dua pola yang berbeda, kapasitas konstan pada A1, dan kompresi bertahap pada A2 sampai A4. Rincian struktur *hidden layer* ditunjukan pada Tabel 2.

Tabel 2. Struktur *hidden layer* MLP

Kode	Jumlah Hidden Layer	Susunan Neuron
A1	1	768→ 768 → C
A2	2	768 → 512 →256→ C
A3	3	768 → 512 → 256 → 128 → C
A4	4	768 → 512 → 256 → 128→ 64 → C

Pada Tabel 2, Jumlah *hidden layer* 2 tidak mencakup *output layer*, dan simbol (C) menunjukkan jumlah kelas keluaran. A1 mempertahankan dimensi *neuron* konstan mengikuti pola *classifier head* standar *baseline*, sehingga perbandingan antara A1 dan *baseline* mencerminkan pengaruh fungsi aktivasi dan *dropout* secara terisolasi dari perubahan struktur *neuron*. Kombinasi A1 dengan fungsi aktivasi Tanh dan *dropout* 0.1 secara struktural dan inisialisasi identik dengan *baseline*, sehingga hasilnya yang sebanding berfungsi sebagai konfirmasi konsistensi implementasi eksperimen. Struktur A2 sampai A4 perlu dibaca berbeda, ketiganya bukan ablasi murni terhadap kedalaman jaringan, melainkan studi gabungan kedalaman dan kapasitas.

Setiap penambahan *hidden layer* pada struktur ini disertai penyusutan jumlah *neuron* secara proposional, mengikuti pola kompresi bertahap yang lazim pada rancangan MLP. Akibatnya, pengaruh kedalaman jaringan tidak dapat dipisahkan sepenuhnya dari pengaruh kapasitas *neuron* pada masing-masing arsitektur. Pemisahan kedua faktor ini secara independen membutuhkan ruang pengujian yang jauh lebih besar, misalnya dengan menjaga jumlah *neuron* tetap konstan di setiap penambahan hidden layer, sehingga jumlah kombinasi konfigurasi yang harus dilatih dan dievaluasi meningkat signifikan. Pertimbangan keterbatasan waktu dan sumber daya komputasi pada penelitian ini membuat desain tersebut belum dapat diakomodasi, sehingga variasi struktur pada penelitian ini tetap diperlakukan sebagai satu kesatuan faktor gabungan. Hal ini merupakan keterbatasan dalam interpretasi hasil perbandingan antar arsitektur, dan pemisahan kedua faktor secara terkontrol menjadi arah yang terbuka untuk penelitian selanjutnya.

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

Setiap struktur diuji dengan empat fungsi aktivasi. Tanh digunakan sebagai aktivasi acuan karena digunakan pada *classifier head baseline*. Fungsi ini menghasilkan keluaran pada rentang [-1,1] dan bersifat simetris terhadap nol, namun dapat mengalami saturasi pada nilai masukan ekstrem (Dubey et al., 2022). Fungsi Tanh dirumuskan sebagai:

$$\text{Tanh}(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad (2)$$

ReLU digunakan sebagai pembanding berbasis fungsi *rectifier*. Nilai positif diteruskan secara *linear*, sedangkan nilai nol dan negatif dipetakan menjadi nol. Aktivasi ini digunakan untuk membandingkan kinerja *classifier head* ketika nilai negatif tidak diteruskan ke *layer* berikutnya (Dubey et al., 2022). Fungsi ReLU dirumuskan sebagai:

$$\text{ReLU}(x) = \max(0, x) \quad (3)$$

ELU bersifat linear pada masukan positif dan memberikan respons halus mendekati batas bawah pada masukan nol atau negatif, sehingga nilai negatif tidak seluruh nya dipotong menjadi nol (Dubey et al., 2022). Fungsi ELU dirumuskan sebagai:

$$ELU(x) = \begin{cases} x & x > 0 \\ \alpha(e^{x} - 1) & x \leq 0 \end{cases} \quad (4)$$

dengan (α) sebagai parameter yang mengatur keluaran pada wilayah negatif. GELU membobotkan asukan berdasarkan fungsi distribusi kumulatif Gausian sehingga nilai negatif tetap dapat berkontribusi pada transfornasi *hidden layer* (Lee, 2023). Fungsi GELU dirumuskan sebagai:

$$\text{GELU}(x) = x \cdot \Phi(x) \quad (5)$$

dengan $\left(\Phi(x)\right)$sebagai fungsi distribusi kumulatif Gaussian standar.

Nilai *dropout* yang diuji adalah 0.1 yang mengikuti nilai dedault pada *classifier head* standar *baseline*, dan 0.3 sebgai nilia regulasi yang lebih kuat dan umum digunakan pada fine-tuning model Transformer (Salehin & Kang, 2023). Kombinasi empat struktur, empat fungsi aktivasi, dan dua nilai *dropout* menghasilkan 32 konfigurasi, yang dievaluasi menggunakan *grid search* agar setiap kandidat dalam ruang pencarian yang bersifat diskret dapat dievaluasi dengan prosedur yang sama (Bischl et al., 2023). *Baseline* dan seluruh konfigurasi MLP dievaluasi menggunakan *Stratified* 5-*Fold Cross-Validation* pada data latih (Szeghalmy & Fazekas, 2023). Pada setiap *fold*, *checkpoint* ditentukan berdasarkan Macro-F1 validasi tertinggi. Kinerja setiap konfigurasi MLP diringkas menggunakan rata-rata Macro-F1 pada lima *fold*. Konfigurasi tersebut kemudian dilatih ulang menggunakan seluruh data latih resmi sebelum digunakan untuk menghasilkan prediksi pada data uji.

2.6 Prosedur Peatihan Model

Baseline dan seluruh konfigurasi MLP dilatih melalui *fine-tuning end-to-end* menggunakan *optimizer* AdamW dengan *learning rate*, 2×10^{-5} untuk RoBERTa sebagai *encoder* dan 2×10^{-4} untuk *classifier head*. Perbedaan *learning rate* ini diterapkan karena encoder RoBERTa menggunakan bobot pra-latih yang telah mengandung representasi linguistik umum sehingga pembaruan yang terlalu besar beresiko merusak representasi tersebut, sedangkan *classifier* diinisialisasi secara acak sehingga membutuhkan pembaruan yang lebih besar di awal pelatihan. *Weigh decay* diterapkan sebesar 0.01 dan *warmup ratio* sebesar 0.06. Laju pembelajaran diatur menggunakan *linear learning rate* dan *gradient clipping* dengan nilai maksimum 1.0 ditetapkan untuk menjaga stabilitas pelatihanan. *Batch size* yang digunakan adalah 32 dan seluruh konfigurasi dilatih selama 4 *epoch*.

2.7 Evaluasi Model

Kinerja model dievaluasi menggunakan Macro-F1 sebagai metrik utama karena setiap kelas memiliki kontribusi yang sama dalam penilaian tanpa mempertimbangkan jumah sampel per kelas, sehingga metri ini lebih sesuai untuk dataset dengan distribusi kelas yang tidak seimbang. Pemilihan Macro-F1 juga konsisten dengan metrik evaluasi resmi HASOC 2021. *Precision*, *recall*, dan F1-*score* untuk kelas ke-i dihitung menggunakan persamaan berikut.

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad (6)$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (7)$$

$$F1_i = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (8)$$

$$Macro\text{-}F1 = \frac{1}{K} \sum_{i=1}^{K} F\,1_i \quad (9)$$

Pada persamaan tersebut, (TP_i) adalah jumlah data kelas ke-i yang diprediksi benar, (FP_i) adalah jumlah data dari kelas lain yang salah diprediksi sebagai kelas ke-i, dan (FN_i) adalah jumlah data kelas ke-(i) yang salah diprediksi sebagai kelas lain, dan (K) adalah jumlah kelas.

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

3. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil evaluasi untuk menganalisis pengaruh konfigurasi MLP sebagai *classifier head* terhadap performa model RoBERTa-*base* pada tugas klasifikasi ujaran kebencian. Evaluasi dilakukan secara bertahap dengan membandingkan model baseline terhadap berbagai kombinasi struktur hidden layer, fungsi aktivasi, dan nilai *dropout* menggunakan *5-fold cross-validation*. Selanjutnya, konfigurasi terbaik dievaluasi pada data uji independen untuk menilai kemampuan generalisasi model, kemudian dibandingkan dengan benchmark HASOC 2021 guna menempatkan hasil penelitian dalam konteks performa metode-metode terdahulu. Selain menyajikan hasil kuantitatif, bagian ini menginterpretasikan setiap temuan berdasarkan karakteristik model, konsistensi performa antar-*fold*, serta implikasinya terhadap efektivitas konfigurasi *classifier head* yang diusulkan.

3.1 Evaluasi Model *Baseline*

Sebelum mengevaluasi berbagai konfigurasi MLP, RoBERTaForSequenceClassification dengan *classifier head* bawaan terlebih dahulu dievaluasi sebagai *baseline* menggunakan prosedur yang sama dengan seluruh konfigurasi yang diusulkan. *Baseline* berfungsi sebagai titik acuan untuk membandingkan performa setiap konfigurasi MLP sehingga pengaruh modifikasi *classifier head* terhadap performa model dapat dievaluasi secara objektif. Hasil evaluasi *baseline* pada kedua subtask disajikan pada Tabel 3.

Tabel 3. Hasil evaluasi *baseline*

Subtask	Macro-F1
Subtask A	0.8038 ± 0.0150
Subtask B	0.6530 ± 0.0175

Berdasarkan Tabel 3, *baseline* memperoleh Macro-F1 yang jauh lebih tinggi pada Subtask A dibandingkan Subtask B, dengan variasi antar-*fold* yang relatif kecil pada kedua *subtask*. Kesenjangan ini konsisten dengan perbedaan struktur masalah pada kedua *subtask*, bukan dengan kemampuan *classifier head* itu sendiri. Subtask A merupakan klasifikasi biner yang memisahkan ujaran bermasalah dari ujaran netral melalui satu batas keputusan, sehingga kesalahan klasifikasi hanya dapat terjadi dalam satu arah per kelas. Subtask B mengharuskan model menentukan batas keputusan pada empat kelas sekaligus, dengan tiga kelas pertama sama-sama memuat bahasa bermuatan negatif namun berbeda pada ada tidaknya sasaran kebencian yang spesifik maupun tingkat kevulgaran kata yang digunakan. Batas antar kelas tersebut jauh lebih kabur dibandingkan batas antara ujaran bermasalah dan ujaran netral pada Subtask A, sehingga *classifier head* menghadapi kesulitan lebih besar dalam memetakan representasi CLS ke logit kelas yang tepat. Dengan demikian, kesenjangan performa pada Tabel 3 lebih tepat dipahami sebagai konsekuensi dari perbedaan struktur ruang label dan distribusi kelas antar-*subtask*, dan bukan sebagai bukti bahwa *classifier head* bawaan bekerja secara inheren lebih baik pada satu *subtask* dibandingkan lainnya.

3.2 Evaluasi Konfigurasi MLP

Bagian ini menyajikan hasil pengujian 32 konfigurasi MLP pada Subtask A dan Subtask B Setiap konfigurasi merupakan kombinasi dari struktur *hidden layer*, fungsi aktivasi, dan nilai *dropout* yang dievaluasi menggunakan *Stratified* 5-*Fold Cross-Validation.* Pengaruh masing-masing komponen dibahas secara terpisah per *subtask* karena pola yang ditemukan berbebda secara fundamental antara klasifikasi biner dan multikelas.

3.2.1 Subtask A

Hasil evaluasi seluruh konfigurasi MLP pada Subtask A disajikan pada Tabel 4. Variasi konfigurasi yang diuji menghasilkan perbedaan performa terhadap *baseline*, meskipun peningkatan yang diperoleh relatif terbatas. Di antara seluruh konfigurasi, A2 dengan fungsi aktivasi ELU dan *dropout* 0.3 memberikan performa validasi tertinggi sehingga dipilih untuk dievaluasi pada data uji. Hasil tersebut menunjukkan bahwa modifikasi *classifier head* tetap mampu mempengaruhi performa RoBERTaForSequenceClassification, namun pengaruh tersebut tidak diperoleh secara merata pada seluruh konfigurasi yang diuji.

Tabel 4. Hasil evakuasi konfigurasi MLP Subtask A

Struktur	*Dropout*	Tanh	ELU	GELU	ReLU
A1	0.1	0.8038 ± 0.0150	0.8034 ± 0.0151	0.8028 ± 0.0172	0.7996 ± 0.0152
	0.3	0.8020 ± 0.0162	0.8046 ± 0.0128	0.8050 ± 0.0155	0.8064 ± 0.0187
A2	0.1	0.8010 ± 0.0151	0.7999 ± 0.0173	0.8001 ± 0.0194	0.7994 ± 0.0156
	0.3	0.8054 ± 0.0156	0.8066 ± 0.0195	0.7995 ± 0.0153	0.8035 ± 0.0171
A3	0.1	0.8024 ± 0.0164	0.8001 ± 0.0205	0.8028 ± 0.0167	0.7965 ± 0.0186
	0.3	0.8029 ± 0.0199	0.7950 ± 0.0165	0.7984 ± 0.0153	0.8023 ± 0.0216
A4	0.1	0.7927 ± 0.0169	0.8020 ± 0.0174	0.8054 ± 0.0192	0.8021 ± 0.0145
	0.3	0.7963 ± 0.0135	0.8015 ± 0.0141	0.8010 ± 0.0182	0.7991 ± 0.0182

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

Hasil pada struktur A1 menunjukkan bahwa perbedaan performa antar struktur hidden layer pada Subtask A berada pada rentang yang seempit, dengan seluruh konfigurasi berkisar di sekitar performa *baseline.* Pola ini sejalan dengan karakteristik Subtask A yang telah diuraikan sebelumnya: batas keputusan biner antara ujaran bermasalah dan ujaran netral relatif lebih mudah dipetakan dari representasi CLS yang dihasilkan *encoder* RoBERTa setelah *fine-tuning* penuh, sehingga penambahan kapasitas maupun kedalaman pada *classifier head* tidak banyak menyumbang informasi baru yang relevan bagi keputusan klasifikasi. Konfigurasi dengan rata-rata Macro-F1 validasi tertinggi diperoleh pada struktur A2, yang mengindikasikan bahwa penambahan satu *hidden layer* disertai kompresi *neuron* secukupnya mampu memberi ruang transformasi tambahan yang mendukung pemisahan kelas, tanpa menimbulkan risiko besar terhadap stabilitas optimasi. Kecenderungan ini, namun demikian, tidak berlanjut secara monoton pada A3 maupun A4, yang menandakan bahwa penambahan kedalaman melampaui titik tertentu mulai memasukkan parameter baru yang harus dipelajari dari kondisi acak tanpa disertai kebutuhan transformasi tambahan yang sepadan, sehingga marjin manfaatnya menyusut seiring bertambahnya kedalaman.

Pengaruh fungsi pada Tabel 4 juga tidak menunjukan keunggulan salah satu fungsi secara konsisten pada seluuruh arsitektur. Kondisi ini dapat dijelaskan dari sifat representasi yang diterima *classifier head*. Representasi CLS keluaran RoBERTa yang telah melalui *fine-tuning* penuh cenderung memiliki rentang nilai yang relatif stabil dan tidak didominasi oleh nilai ekstrem, sehingga perbedaan karakteristik nonlinearitas antar fungsi aktivasi, seperti kecenderungan ReLU memetakan seluruh nilai negatif menjadi nol atau kecenderungan Tanh mengalami saturasi pada nilai masukan ekstrem, tidak banyak berpengaruh ketika masukan yang diterima sudah berada pada rentang yang wajar. Efektivitas suatu fungsi aktivasi dengan demikian lebih ditentukan oleh interaksinya dengan struktur dan nilai *dropout* pada konfigurasi yang sama, bukan oleh karakteristik intrinsik fungsi tersebut semata.

Nilai *dropout* menunjukkan pola serupa, tanpa satu nilai yang secara konsisten unggul pada seluruh konfigurasi. Mengingat *classifier head* pada Subtask A hanya perlu mempelajari satu batas keputusan yang relatif tegas, kebutuhan regularisasi tambahan melalui *dropout* yang lebih besar tidak selalu diperlukan. Pada struktur dengan kapasitas *neuron* yang sudah minim seperti A4, nilai *dropout* yang lebih besar justru berpotensi menghambat pembelajaran karena mengurangi proporsi *neuron* aktif pada saat kapasitas jaringan tersisa sudah terbatas. Efektivitas *dropout* dengan demikian bergantung pada keseimbangan antara kapasitas struktur yang tersedia dan derajat regularisasi yang diterapkan, bukan pada nilai *dropout* itu sendiri secara terpisah.

Secara keseluruhan, hasil pada Tabel 4 menunjukkan bahwa modifikasi *classifier head* mampu memengaruhi performa pada Subtaks A, namun peningkatan tersebut bersifat marjinal dan bersumber dari satu komponen yang mendominasi. Konfigurasi A2 dengan fungsi aktivasi ELU dan *dropout* 0.3 hanya dapat disebut sebagai konfigurasi dengan rata-rata Macro-F1 validasi tertinggi di antara 32 kandidat yang diuji, bukan sebagai konfigurasi yang secara definitif terbukti paling unggul, karena selisihnya terhadap sejumlah konfigurasi lain maupun terhadap *baseline* berada pada rentang yang sempit dan sebanding dengan besarnya variasi antar-*fold* yang tercatat pada Tabel 4 itu sendiri.

3.2.2 Subtask B

Hasil evaluasi seluruh konfigurasi MLP pada Subtask B disajikan pada Tabel 5. Variasi konfigurasi yang dievaluasi menghasilkan perbedaan performa yang lebih nyata dibandingkan Subtask A. Di antara seluruh konfigurasi yang diuji, A1 dengan fungsi aktivasi GELU dan *dropout* 0,1 menghasilkan performa validasi tertinggi sehingga dipilih untuk tahap evaluasi pada data uji. Temuan tersebut menunjukkan bahwa peningkatan performa tidak selalu memerlukan arsitektur *classifier head* yang lebih dalam, tetapi bergantung pada kesesuaian kombinasi struktur, fungsi aktivasi, dan *dropout*.

Tabel 5. Hasil evaluasi konfigurasi MLP Subtask B

Struktur	*Dropout*	Tanh	ELU	GELU	ReLU
A1	0.1	0.6530 ± 0.0175	0.6559 ± 0.0177	0.6586 ± 0.0137	0.6540 ± 0.0137
	0.3	0.6553 ± 0.0170	0.6533 ± 0.0214	0.6508 ± 0.0165	0.6555 ± 0.0185
A2	0.1	0.6509 ± 0.0064	0.6538 ± 0.0078	0.6487 ± 0.0046	0.6447 ± 0.0077
	0.3	0.6435 ± 0.0074	0.6434 ± 0.0134	0.6401 ± 0.0117	0.6409 ± 0.0140
A3	0.1	0.6298 ± 0.0229	0.6338 ± 0.0178	0.6319 ± 0.0153	0.6300 ± 0.0180
	0.3	0.6334 ± 0.0148	0.6243 ± 0.0248	0.6231 ± 0.0167	0.6154 ± 0.0275
A4	0.1	0.5372 ± 0.0297	0.5623 ± 0.0274	0.4918 ± 0.0018	0.4984 ± 0.0149
	0.3	0.5139 ± 0.0172	0.5364 ± 0.0408	0.3702 ± 0.0502	0.4486 ± 0.0569

Pola pada Tabel 5 memperlihatkan kecenderungan yang berbeda secara mendasar dari Subtask A. Struktur A1 dan A2 masih mempertahankan performa yang sebanding dengan *baseline*, tetapi performa menurun pada A3 dan menurun tajam pada A4 di seluruh kombinasi fungsi aktivasi dan *dropout* yang diuji. Penurunan yang konsisten pada seluruh sel A4, alih-alih hanya pada sebagian kombinasi tertentu, mengindikasikan bahwa sumber masalah bukan terletak pada pemilihan fungsi aktivasi atau *dropout* tertentu, melainkan pada struktur itu sendiri, yaitu empat *hidden layer* dengan kompresi neuron bertahap dari 768 hingga 64. Sifat sistematis dari penurunan ini, yang terjadi pada seluruh kombinasi hyperparameter lain, menjadi dasar bagi analisis error yang lebih mendalam pada subbab berikut.

Temuan yang paling menonjol pada Subtask B muncul pada struktur A4. Dibandingkan struktur lainnya, A4 tidak hanya menghasilkan Macro-F1 terendah, tetapi juga memperlihatkan variasi performa antar *fold* yang lebih besar

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

pada sebagian besar konfigurasi. Kombinasi kedua indikator tersebut mengisyaratkan bahwa konfigurasi dengan kedalaman jaringan tertinggi pada penelitian ini belum mampu mempertahankan performa maupun konsistensi hasil selama validasi silang. Menariknya, salah satu konfigurasi A4 menghasilkan standar deviasi yang sangat kecil meskipun nilai Macro-F1 tetap rendah. Hal ini menunjukkan bahwa performa yang rendah tersebut muncul secara konsisten pada seluruh *fold*, sehingga konsistensi hasil tidak selalu diikuti oleh kualitas prediksi yang lebih baik. Oleh karena itu, interpretasi terhadap standar deviasi perlu dilakukan bersama nilai Macro-F1 agar evaluasi tidak hanya berfokus pada konsistensi hasil, tetapi juga mempertimbangkan kualitas performa yang dicapai.

Degradasi performa pada struktur A4 di Subtask B, sebagaimana ditunjukkan pada Tabel 5, dapat ditelusuri melalui dua kemungkinan penjelasan yang bersifat berlawanan arah: *overfitting* yang parah terhadap data latih, atau hilangnya informasi kelas yang relevan akibat kompresi representasi yang terlalu agresif pada *classifier head*. Kedua kemungkinan ini dapat dibedakan melalui pola yang muncul ketika nilai *dropout* dinaikkan dari 0.1 menjadi 0.3. Apabila penyebab utamanya adalah *overfitting*, penguatan regularisasi melalui *dropout* yang lebih besar seharusnya membantu memperbaiki performa validasi, karena *dropout* secara langsung berfungsi menekan kapasitas efektif jaringan agar tidak menghafal pola data latih. Pada Tabel 5, kondisi yang teramati justru sebaliknya, peningkatan *dropout* pada A4 diikuti oleh penurunan performa lebih lanjut pada hampir seluruh fungsi aktivasi, dengan penurunan paling ekstrem terjadi pada kombinasi GELU dan ReLU. Pola ini tidak sejalan dengan penjelasan *overfitting* semata, karena regularisasi tambahan semestinya meringankan gejala tersebut, bukan memperburuknya.

Pola yang teramati lebih konsisten dengan penjelasan hilangnya informasi akibat kompresi representasi yang terlalu agresif. Struktur A4 menyusutkan representasi 768 dimensi keluaran *encoder* melalui empat lapisan tersembunyi secara bertahap menuju 64 *neuron* sebelum mencapai lapisan keluaran, jauh lebih agresif dibandingkan A1 hingga A3. Seluruh lapisan pada *classifier head* diinisialisasi secara acak dan dilatih bersama *encoder* yang telah memiliki representasi linguistik matang dari pra-pelatihan. Ketika empat transformasi non-linear yang belum terlatih disusun berurutan dengan penyempitan dimensi yang tajam, sinyal gradien yang merambat kembali ke *encoder* selama pelatihan awal berisiko menjadi tidak informatif, sementara pada arah maju, informasi pembeda antar kelas yang tumpang tindih pada Subtask B, berisiko tereduksi sebelum sempat dipelajari secara memadai oleh lapisan-lapisan awal yang masih acak. Penambahan *dropout* pada kondisi ini menghapus lebih banyak *neuron* aktif dari jaringan yang kapasitasnya sudah tereduksi tajam, sehingga informasi yang tersisa untuk membedakan kelas menjadi semakin sedikit dan performa semakin menurun alih-alih membaik. Standar deviasi antar-*fold* yang lebih besar pada sebagian konfigurasi A4 di Tabel 5 turut memperkuat penjelasan ini, karena jaringan dengan kapasitas informasi yang sudah tereduksi cenderung lebih sensitif terhadap variasi kecil pada inisialisasi bobot maupun komposisi data latih tiap *fold*, sehingga hasilnya menjadi tidak stabil. Dengan mempertimbangkan kedua indikasi tersebut, penurunan performa pada struktur A4 lebih tepat diatribusikan pada hilangnya informasi akibat kompresi representasi yang berlebihan pada *classifier head* yang belum terlatih, dan bukan pada *overfitting* terhadap data latih. Perlu dicatat bahwa atribusi ini didasarkan pada pola tidak langsung yang teramati dari respons performa terhadap perubahan *dropout* dan besarnya variasi antar-*fold*, bukan dari perbandingan langsung antara *loss* pelatihan dan *loss* validasi, sehingga kepastian penuh mengenai mekanisme yang terjadi memerlukan diagnosis tambahan seperti pemantauan kurva pelatihan pada penelitian selanjutnya.

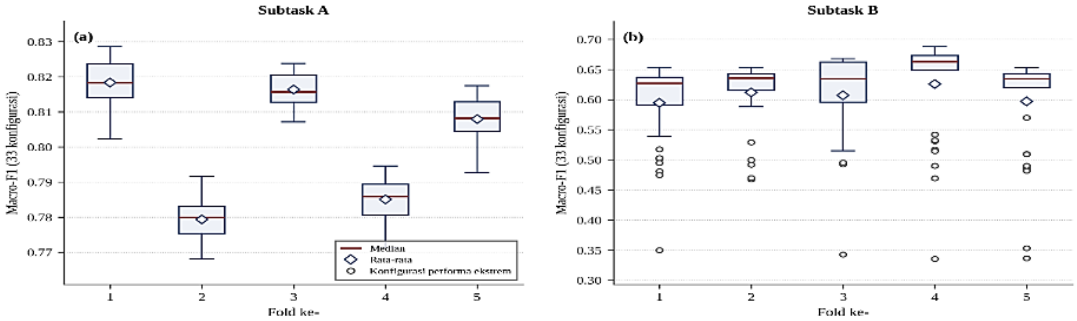
Variasi fungsi aktivasi dan *dropout* pada struktur A1 hingga A3 juga tidak memperlihatkan adanya komponen yang secara konsisten mendominasi seluruh konfigurasi. GELU menghasilkan konfigurasi dengan performa validasi tertinggi pada A1, namun keunggulan tersebut tidak bertahan pada struktur lainnya, menunjukkan bahwa keunggulan suatu fungsi aktivasi bergantung pada kesesuaiannya dengan struktur tempat fungsi tersebut diterapkan, bukan pada sifat intrinsik fungsi itu sendiri. Nilai *dropout* yang lebih besar juga tidak secara konsisten memperbaiki performa, sebuah pola yang selaras dengan penjelasan pada struktur A4: efektivitas *dropout* bergantung pada apakah kapasitas struktur yang tersisa masih memadai untuk menampung informasi kelas setelah sebagian neuron dinonaktifkan.

Secara keseluruhan, hasil Subtask B menunjukkan bahwa efektivitas *classifier head* tidak ditentukan oleh peningkatan kompleksitas arsitektur semata, melainkan oleh kesesuaian antara kapasitas struktur, fungsi aktivasi, dan *dropout* dengan karakteristik tugas yang dihadapi. Konfigurasi A1 dengan GELU dan *dropout* 0,1 hanya unggul tipis dibandingkan *baseline* pada tahap validasi, sehingga lebih tepat dipahami sebagai konfigurasi dengan rata-rata Macro-F1 tertinggi di antara kandidat yang diuji, bukan sebagai bukti bahwa modifikasi *classifier head* secara definitif lebih baik daripada konfigurasi bawaan. Penurunan performa yang tajam disertai meningkatnya variasi hasil pada A4 memperlihatkan bahwa penambahan kompleksitas melewati titik tertentu memberikan kerugian yang jauh lebih nyata dibandingkan potensi keuntungannya pada tugas klasifikasi multikelas ini.

3.3 Analisis Variasi Antar Fold

Nilai rata-rata Macro-F1 pada Tabel 4 dan Tabel 5 merangkum performa lima *fold* menjadi satu angka, sehingga pola ketidakstabilan yang terjadi pada *fold* tertentu tidak tampak secara eksplisit. Subbab ini menelaah sebaran Macro-F1 dari seluruh konfigurasi pada setiap *fold* untuk kedua *subtask*, guna mengidentifikasi apakah ketidakstabilan performa bersumber dari karakteristik *fold* tertentu atau dari konfigurasi *classifier head* yang diuji.

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512



Gambar 3. Rata-rata Macro-F1 per-fold

Gambar 3 menunjukkan bahwa pada Subtask A, *fold* ke-2 dan *fold* ke-4 secara konsisten menghasilkan Macro-F1 yang lebih rendah dibandingkan *fold* ke-1, ke-3, dan ke-5, dan pola ini berlaku hampir merata pada seluruh konfigurasi yang dievaluasi, termasuk *baseline*. Karena *baseline* tidak melibatkan modifikasi apapun pada *classifier head*, penurunan performa yang tetap muncul pada *fold* yang sama mengindikasikan bahwa sumber ketidakstabilan ini lebih mungkin berasal dari karakteristik data pada partisi *fold* ke-2 dan ke-4, misalnya proporsi sampel dengan ujaran implisit, sarkasme, atau bahasa berkode yang lebih tinggi pada partisi tersebut, dibandingkan dari kelemahan arsitektur *classifier head* itu sendiri. Stratifikasi lima lipatan pada penelitian ini menjaga proporsi label tetap seimbang antar-*fold*, tetapi tidak mengendalikan tingkat kesulitan linguistik sampel di dalamnya, sehingga *fold* dengan proporsi kasus ambigu yang lebih tinggi tetap dapat menghasilkan performa yang lebih rendah meskipun distribusi kelasnya sebanding dengan *fold* lain.

Subtask B memperlihatkan pola yang berbeda dan tidak sejalan dengan Subtask A. Alih-alih ikut menurun pada *fold* ke-2 dan ke-4 sebagaimana pada Subtask A, Gambar 3 menunjukkan bahwa *fold* ke-4 pada Subtask B justru menghasilkan sebaran Macro-F1 tertinggi di antara kelima *fold*, sedangkan *fold* ke-1 cenderung menghasilkan performa terendah. Anomali ini mengindikasikan bahwa kesulitan yang dialami suatu *fold* pada satu *subtask* tidak dapat diasumsikan berlaku sama pada *subtask* lain, meskipun kedua *subtask* berasal dari partisi data yang sama. Perbedaan ini masuk akal mengingat Subtask B melibatkan empat kelas dengan proporsi yang jauh lebih timpang, sehingga komposisi kelas minoritas seperti HATE pada tiap *fold* jauh lebih menentukan performa dibandingkan pada Subtask A yang hanya membedakan dua kelas. Sebuah *fold* yang kebetulan memuat proporsi sampel HATE atau OFFN dengan pola bahasa yang relatif lebih konsisten dapat menghasilkan performa yang justru lebih baik pada Subtask B, meskipun *fold* yang sama menyulitkan Subtask A karena alasan yang berbeda, misalnya tingkat sarkasme yang lebih tinggi tanpa memandang kategori multikelasnya. Perlu dicatat pula bahwa seluruh *fold* pada Subtask B memuat titik sebaran ekstrem bernilai rendah secara konsisten, sebagaimana tampak pada Gambar 3(b). Titik-titik tersebut bersesuaian dengan konfigurasi berstruktur A4 yang telah dibahas pada subbab Analisis Error, dan kemunculannya yang merata pada seluruh *fold* alih-alih terkonsentrasi pada *fold* tertentu memperkuat simpulan bahwa penurunan performa A4 bersumber dari struktur *classifier head* itu sendiri, bukan dari komposisi data pada *fold* tertentu.

3.4 Perbandingan Performa Baseline dan Konfigurasi Terbaik

Konfigurasi *classifier head* dengan rata-rata Macro-F1 validasi tertinggi pada masing-masing *subtask* selanjutnya dilatih ulang menggunakan seluruh data latih resmi dan dievaluasi pada data uji independen untuk dibandingkan dengan model *baseline*. Perbandingan dilakukan menggunakan metrik *precision*, *recall*, dan Macro-F1. . Hasil pengujian disajikan pada Tabel 6.

Tabel 6. Perbandingan pada data uji

Subtask	Model	precision	Recall	Macro-F1
A	*Baseline*	0.8159	0.7995	0.8059
	A2 ELU 0.3	0.8275	0.7994	0.8090
B	*Baseline*	0.6589	0.6638	0.6611
	A1 GELU 0.1	0.6543	0.6602	0.6568

Pada Subtask A, konfigurasi A2 tetap menghasilkan Macro-F1 yang lebih tinggi dibandingkan *baseline* pada Tabel 6, konsisten dengan keunggulannya selama tahap *cross-validation*. Peningkatan *precision* menjadi kontributor utama pada hasil ini, sementara *recall* kedua model relatif setara, yang menunjukkan bahwa struktur A2 sedikit memperbaiki ketegasan batas keputusan tanpa mengorbankan kemampuan menangkap kelas positif. Meskipun demikian, peningkatan yang diperoleh tetap tergolong kecil, sehingga hasil ini lebih tepat dibaca sebagai indikasi bahwa modifikasi *classifier head* dapat memberikan manfaat tambahan yang tipis pada Subtask A, bukan sebagai bukti keunggulan yang besar.

Pola yang berbeda muncul pada Subtask B. Konfigurasi A1 GELU 0,1, yang menghasilkan rata-rata Macro-F1 validasi tertinggi selama *cross-validation*, tidak lagi mengungguli *baseline* pada data uji sebagaimana ditunjukkan pada Tabel 6. Temuan ini bukan sekadar hasil kebetulan pada satu titik data, melainkan dapat dijelaskan melalui dua faktor

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

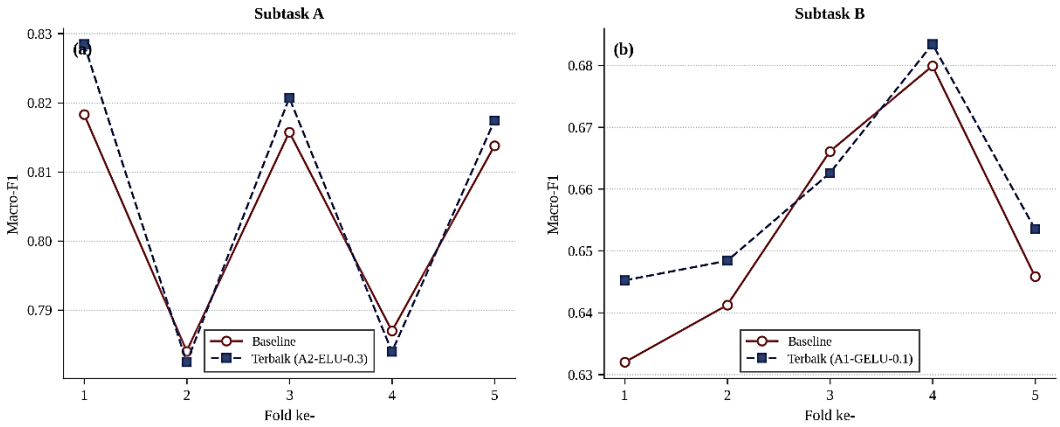
yang saling berkaitan. Pertama, proses pemilihan konfigurasi terbaik dilakukan dengan membandingkan rata-rata validasi dari 32 kandidat sekaligus. Ketika sejumlah besar kandidat dibandingkan pada rentang performa yang sempit sebagaimana terlihat pada Tabel 5, konfigurasi dengan rata-rata tertinggi cenderung sebagian terpilih karena kombinasi *fold* yang kebetulan menguntungkan, bukan semata-mata karena keunggulan arsitektur yang konsisten. Kedua, selisih Macro-F1 antara A1 GELU 0.1 dan *baseline* selama *cross-validation* berada pada rentang yang jauh lebih kecil dibandingkan standar deviasi antar-*fold* yang dilaporkan pada Tabel 5, sehingga keunggulan tersebut secara praktis berada dalam rentang variasi normal, bukan keunggulan yang besar dan stabil. Kombinasi kedua faktor ini menjelaskan mengapa urutan performa yang teramati selama *cross-validation* tidak selalu bertahan ketika diuji pada data uji independen yang tidak terlibat dalam proses pemilihan model.

Secara keseluruhan, hasil pada Tabel 6 memperlihatkan perbedaan pola antara kedua subtask. Konfigurasi tertinggi pada Subtask A berhasil mempertahankan keunggulannya hingga tahap pengujian, sedangkan konfigurasi tertinggi pada Subtask B tidak lagi mengungguli model *baseline*. Dengan demikian, apakah modifikasi *classifier head* memberikan keuntungan nyata pada tugas klasifikasi multikelas seperti Subtask B masih menjadi pertanyaan terbuka yang memerlukan pengujian pada data yang lebih besar untuk disimpulkan secara pasti. Temuan ini menegaskan bahwa evaluasi menggunakan data uji independen tetap diperlukan untuk memverifikasi performa akhir model, dan bahwa rata-rata performa *cross-validation* semata tidak cukup untuk menetapkan suatu konfigurasi sebagai konfigurasi terbaik secara definitif.

Tabel 7. Detail performa Subtask B per-kelas

	Kelas	Precision	Recall	F1-Score	Support
Baseline	HATE	0.5630	0.5982	0.5801	224
	NONE	0.7717	0.7350	0.7529	483
	OFFN	0.5327	0.5436	0.5381	195
	PRFN	0.7682	0.7784	0.7733	379
A1 GELU 01	HATE	0.5630	0.5982	0.5801	224
	NONE	0.7879	0.7308	0.7583	483
	OFFN	0.5124	0.5282	0.5202	195
	PRFN	0.7538	0.7836	0.7684	379

Tabel 7 menguraikan performa per kelas pada Subtask B untuk menelusuri sumber selisih Macro-F1 antara *baseline* dan A1 GELU 0.1 yang tercatat pada Tabel 6. Pada kelas HATE, kedua model menghasilkan *precision*, *recall*, dan F1-*score* yang identik, menunjukkan bahwa kedua *classifier head* menghasilkan pola klasifikasi yang sama persis untuk kelas ini. Pada kelas NONE, A1 GELU 0.1 unggul tipis dibandingkan *baseline*, dengan *precision* yang lebih tinggi namun *recall* yang sedikit lebih rendah, yang berarti model ini lebih ketat dalam menandai suatu *tweet* sebagai NONE namun sedikit lebih sering meloloskan *tweet* NONE sebagai kelas lain. Selisih yang menentukan justru muncul pada kelas OFFN, tempat *baseline* mengungguli A1 GELU 0.1 pada *precision* maupun *recall* sekaligus, sehingga menghasilkan F1-*score* yang lebih tinggi. Kelas PRFN juga sedikit lebih baik pada *baseline*, meskipun selisihnya lebih kecil dibandingkan kelas OFFN. Mengingat kelas OFFN merupakan kelas dengan jumlah sampel uji kedua tersedikit setelah HATE dan secara semantik paling mudah tumpang tindih dengan HATE maupun PRFN, sebagaimana telah diuraikan pada pembahasan Tabel 3, penurunan performa pada kelas inilah yang paling banyak berkontribusi terhadap posisi A1 GELU 0.1 yang berada di bawah *baseline* pada Macro-F1 keseluruhan di Tabel 6, meskipun konfigurasi tersebut unggul pada kelas NONE yang berjumlah sampel lebih banyak.



Gambar 4. Perbandingan model pada variasi antar-fold

Gambar 4 membandingkan Macro-F1 *baseline* dan konfigurasi terbaik pada masing-masing *fold* selama *cross-validation*, sebelum kedua model dilatih ulang untuk evaluasi akhir pada Tabel 6. Pada Subtask A, kedua model bergerak hampir berimpit pada seluruh lima *fold*, termasuk turut menurun bersama pada *fold* ke-2 dan ke-4 sebagaimana

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

telah diuraikan pada Gambar 3, yang menguatkan simpulan bahwa penurunan performa pada kedua *fold* tersebut bersumber dari karakteristik data, bukan dari perbedaan arsitektur *classifier head*. Pada Subtask B, konfigurasi A1 GELU 0.1 berada tipis di atas *baseline* pada seluruh lima *fold*, namun jarak keunggulannya tetap kecil dan tidak melebar pada *fold* manapun. Konsistensi arah keunggulan tanpa disertai besarnya selisih yang meyakinkan ini sejalan dengan penjelasan pada pembahasan Tabel 6 mengenai mengapa keunggulan yang tampak stabil selama *cross-validation* tetap dapat terbalik ketika diuji pada data uji independen: keunggulannya konsisten arah, tetapi konsisten kecil, sehingga pergeseran sekecil apapun pada distribusi data uji sudah cukup untuk mengubah urutan performa akhir.

3.5 Evaluasi Terhadap *Benchmark* HASOC 2021

Untuk menempatkan performa model yang diusulkan dalam konteks penelitian sebelumnya, model dengan performa tertinggi pada masing-masing *subtask* dibandingkan dengan beberapa model berperingkat teratas yang dilaporkan pada kompetisi HASOC menggunakan metrik Macro-F1. Hasil perbandingan tersebut disajikan pada Tabel 8.

Tabel 8. Perbandingan performa model terhadap benchmark HASOC 2021

Subtask	rank	Sistem	Model	Macro-F1
A	1	NLP-CIC	RoBERTa-tw-large	0.8305
	2	HUNLP	GCN	0.8215
	3	neuro-utmn-thales	Twitter-RoBERTa	0.8199
		Penelitian ini	RoBERTa-base + A2 ELU 0.3	0.8090
B	1	NLP-CIC	RoBERTa-tw-large	0.6657
	2	neuro-utmn-thales	Twitter-RoBERTa	0.6577
	3	HASOC21rub	BERTweet-large	0.6482
		Penelitian ini	RoBERTa-base+A1 GELU 0.1	0.6568

Berdasarkan Tabel 8, model yang diusulkan pada Subtask A memperoleh nilai Macro-F1 yang berada di bawah tiga model teratas pada HASOC. Hasil tersebut menunjukkan posisi performa model yang diusulkan terhadap pendekatan-pendekatan dengan performa tertinggi yang dilaporkan pada benchmark HASOC.

Pada Subtask B, model yang diusulkan memperoleh nilai Macro-F1 yang lebih tinggi dibandingkan salah satu model pada tiga besar HASOC, dan selisih performanya terhadap model dengan Macro-F1 tertinggi juga tergolong kecil. Perlu dicatat bahwa pada Subtask B, model *baseline* penelitian ini, yaitu *classifier head* bawaan RoBERTaForSequenceClassification tanpa modifikasi apa pun, justru memperoleh Macro-F1 lebih tinggi daripada konfigurasi terbaik hasil eksperimen sendiri, serta mendekati capaian sistem peringkat pertama. Dengan kata lain, pada Subtask B posisi terbaik penelitian ini terhadap benchmark HASOC sebenarnya dicapai oleh model *baseline*, bukan oleh konfigurasi *classifier head* hasil eksperimen. Hasil tersebut menunjukkan bahwa model yang diusulkan, termasuk model *baseline* itu sendiri, mampu memperoleh posisi yang baik di antara model-model yang dilaporkan pada benchmark HASOC yang rata-rata menggunakan model berbasis *large* untuk Subtask B..

Secara keseluruhan, evaluasi terhadap benchmark HASOC menunjukkan bahwa model yang diusulkan memperoleh performa yang berada pada tingkat yang sebanding dengan hasil yang dilaporkan oleh beberapa model terbaik HASOC pada kedua subtask. Hasil ini memberikan gambaran mengenai posisi model yang diusulkan dalam konteks penelitian sebelumnya serta menunjukkan bahwa pendekatan yang digunakan mampu menghasilkan performa yang layak diperhitungkan pada benchmark HASOC.

3.6 Pembahasan

Sebagaimana telah diuraikan pada bagian Pendahuluan, penelitian-penelitian terdahulu pada HASOC 2021 menempuh pendekatan yang beragam: Bölücü & Canbay (2021) menggunakan arsitektur berbasis *graf* di luar kerangka Transformer, Glazkova et al. (2021) mengadaptasi backbone Transformer ke domain Twitter, Mitra & Sankhala (2021) memodifikasi fungsi loss, sedangkan Kui (2021) serta Agustian et al. (2021) menambahkan lapisan pemrosesan sekuensial setelah representasi Transformer dihasilkan. Penelitian ini menempuh arah yang berbeda dari kelima pendekatan tersebut, *encoder* dan fungsi *loss* dipertahankan pada kondisi standar, sedangkan variasi difokuskan hanya pada *classifier head*.

Konfigurasi terbaik penelitian ini memperoleh Macro-F1 sebesar 0.8090 pada Subtask A dan 0.6568 pada Subtask B. Pada Subtask A, hasil ini sedikit lebih tinggi dibandingkan Mitra & Sankhala (0.8089), tetapi berada di bawah Bölücü & Canbay (0.8215) maupun Glazkova et al. (0.8199). Kesamaan skor dengan Mitra & Sankhala menarik untuk dicermati: penelitian tersebut memodifikasi fungsi *loss* pada BERT-*large*, sementara penelitian ini hanya memodifikasi *classifier head* pada RoBERTa-*base* yang berukuran lebih kecil, namun keduanya berujung pada performa yang hampir identik. Kesamaan ini bersifat indikatif dan tidak dapat dijadikan bukti sebab-akibat, mengingat kedua penelitian berbeda pada banyak aspek sekaligus. Pada Subtask B, hasil penelitian ini melampaui salah satu sistem tiga besar HASOC 2021 BERTweet-*large*, 0.6482 dan mendekati capaian Glazkova et al. (0.6577), dengan selisih yang tergolong tipis. Menariknya, model *baseline* penelitian ini, tanpa modifikasi apa pun pada *classifier head*, justru memperoleh Macro-F1 sebesar 0.6611 pada data uji Subtask B, lebih tinggi daripada konfigurasi terbaik hasil eksperimen sendiri (0.6568), dan melewati capaian Glazkova et al. (0.6577). Temuan ini menunjukkan bahwa *classifier*

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

head bawaan RoBERTaForSequenceClassification, tanpa modifikasi, sudah cukup kompetitif terhadap sejumlah sistem benchmark HASOC 2021 pada tugas multikelas.

Penelitian ini tidak dapat membandingkan besaran peningkatan performa yang diperoleh masing-masing penelitian terdahulu terhadap *baseline* mereka sendiri, karena informasi tersebut umumnya tidak dilaporkan secara eksplisit pada publikasi yang dikutip. Perbandingan pada penelitian ini karenanya terbatas pada posisi skor akhir, bukan pada besarnya kontribusi tiap komponen yang dimodifikasi oleh masing-masing pendekatan. Yang dapat disimpulkan langsung dari penelitian ini sendiri adalah bahwa peningkatan performa melalui modifikasi classifier head bersifat kecil, dan pada Subtask B tidak bertahan pada data uji independen. Temuan ini menunjukkan bahwa modifikasi pada classifier head saja, tanpa menyentuh encoder maupun fungsi loss, memberi ruang perbaikan yang terbatas pada skenario *full fine-tuning*. Hal ini tidak berarti konfigurasi classifier head tidak perlu dipertimbangkan, mengingat penelitian ini tetap menunjukkan bahwa pemilihan struktur yang tidak tepat, seperti pada A4 di Subtask B yang Macro-F1-nya turun hingga ke kisaran 0.37–0.56 pada sejumlah kombinasi fungsi aktivasi dan dropout, dapat merusak performa secara nyata. Dengan demikian, evaluasi konfigurasi classifier head tetap relevan dilakukan terutama untuk menghindari pilihan yang merugikan.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa konfigurasi internal *classifier head* bawaan pada RoBERTaForSequenceClassification memengaruhi performa klasifikasi *hate speech* dan *offensive language* dalam skenario *full fine-tuning*. Pengaruh tersebut tidak ditentukan oleh satu komponen secara terpisah, melainkan oleh interaksi antara struktur *hidden layer*, fungsi aktivasi, dan nilai *dropout* yang membentuk konfigurasi *classifier head* secara keseluruhan. Kontribusi utama penelitian ini terletak pada evaluasi terkontrol dan sistematis terhadap konfigurasi tersebut dalam kondisi *full fine-tuning*, sebuah aspek yang belum banyak dikaji pada penelitian klasifikasi *hate speech* sebelumnya, sekaligus penyediaan bukti empiris dari data uji independen bahwa peningkatan kompleksitas arsitektur *classifier head* tidak secara otomatis menghasilkan performa yang lebih baik. Efektivitas suatu konfigurasi justru bergantung pada kesesuaiannya dengan karakteristik tugas klasifikasi yang dihadapi, dan konfigurasi yang memberikan hasil terbaik pada satu *subtask* belum tentu memberikan kecenderungan yang sama pada *subtask* lainnya. Mengingat pengujian pada penelitian ini hanya dilakukan pada satu dataset dan satu varian *backbone*, temuan berikut sebaiknya dibaca sebagai pedoman awal yang perlu diverifikasi ulang pada konteks lain, bukan sebagai aturan yang berlaku universal bagi seluruh tugas klasifikasi teks. Pada tugas klasifikasi biner atau tugas dengan batas antar kelas yang relatif tegas, *classifier head* bawaan tampak sudah cukup memadai sebagai titik awal, sehingga penambahan *hidden layer*, jika hendak dicoba, cenderung lebih aman dibatasi pada satu hingga dua lapisan dengan kompresi *neuron* yang moderat, karena penambahan lapisan lebih lanjut pada penelitian ini tidak memberi manfaat yang sepadan dengan risikonya. Pada tugas klasifikasi multikelas dengan batas antar kelas yang tumpang tindih, praktisi perlu berhati-hati terhadap struktur *classifier head* yang dalam disertai kompresi dimensi yang agresif, karena kombinasi tersebut pada penelitian ini terbukti dapat merusak performa secara nyata, dan tetap menyertakan *classifier head* bawaan sebagai pembanding alih-alih berasumsi bahwa modifikasi apa pun akan memberi perbaikan. Praktisi juga disarankan untuk memverifikasi konfigurasi terpilih pada data uji independen yang terpisah dari proses pemilihan model, mengingat penelitian ini menemukan bahwa keunggulan yang tampak konsisten selama validasi silang tidak selalu bertahan pada data uji, terutama ketika selisih performa antar-konfigurasi tergolong tipis. Dengan demikian, penelitian ini menawarkan indikasi awal bahwa evaluasi sistematis terhadap konfigurasi *classifier head* tetap layak dipertimbangkan sebagai bagian dari proses perancangan model, sekalipun manfaatnya bersifat marjinal dan bergantung pada tugas, sehingga melengkapi penelitian-penelitian sebelumnya yang lebih banyak berfokus pada pengembangan *backbone*, modifikasi fungsi *loss*, atau penggantian arsitektur klasifikasi secara keseluruhan. Penelitian ini memiliki keterbatasan karena variasi struktur MLP yang diuji mengubah jumlah *hidden layer* dan jumlah *neuron* secara bersamaan, sehingga pengaruh masing-masing faktor belum dapat dipisahkan secara independen. Selain itu, evaluasi hanya dilakukan menggunakan dataset HASOC 2021 bahasa Inggris dengan *backbone* RoBERTa-*base*, sehingga generalisasi temuan terhadap dataset, bahasa, maupun model Transformer lain masih memerlukan kajian lebih lanjut. Oleh karena itu, penelitian selanjutnya dapat mengevaluasi setiap komponen *classifier head* secara lebih terisolasi, memperluas pengujian pada berbagai dataset dan *backbone* yang berbeda, serta mengkaji konfigurasi *classifier head* yang lebih beragam untuk memperoleh pemahaman yang lebih komprehensif mengenai perannya dalam klasifikasi teks.

REFERENCES

Agustian, S., Saputra, R., & Fadhilah, A. (2021). Feature Selection with Pretrained-BERT for Hate Speech and Offensive Content Identification in English and Hindi Languages. *Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December 13-17, 2021, CEUR Workshop Proceedings, 3159*, 508–516. https://ceur-ws.org/Vol-3159/T1-50.pdf

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A., Deng, D., & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery*, *13*(2), e1484. https://doi.org/10.1002/widm.1484

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

Bölücü, N., & Canbay, P. (2021). Hate Speech and Offensive Content Identification with Graph Convolutional Networks. *Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December 13-17, 2021, CEUR Workshop Proceedings*, *3159*, 44–51. https://ceur-ws.org/Vol-3159/T1-4.pdf

Cinelli, M., Pelicon, A., Mozetič, I., Quattrociocchi, W., Novak, P. K., & Zollo, F. (2021). Dynamics of online hate and misinformation. *Scientific Reports*, *11*(1), 22083. https://doi.org/10.1038/s41598-021-01487-w

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022a). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, *503*, 92–108. https://doi.org/10.1016/j.neucom.2022.06.111

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022b). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, *503*, 92–108. https://doi.org/https://doi.org/10.1016/j.neucom.2022.06.111

ElSherief, M., Ziems, C., Muchlinski, D., Anupindi, V., Seybolt, J., De Choudhury, M., & Yang, D. (2021). Latent Hatred: A Benchmark for Understanding Implicit Hate Speech. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 345–363. https://doi.org/10.18653/v1/2021.emnlp-main.29

Glazkova, A. V, Kadantsev, M., & Glazkov, M. (2021). Fine-tuning of Pre-trained Transformers for Hate Offensive and Profane Content Detection in English and Marathi. *Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December 13-17, 2021, CEUR Workshop Proceedings*, *3159*, 52–62. https://ceur-ws.org/Vol-3159/T1-5.pdf

Kenneweg, P., Schröder, S., & Hammer, B. (2022). Neural Architecture Search for Sentence Classification with BERT. *ESANN 2022 Proceedings*, 417–422. https://doi.org/10.14428/esann/2022.ES2022-45

Kui, Y. (2021). Detect Hate and Offensive Content in English and Indo-Aryan Languages based on Transformer. *Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December 13-17, 2021, CEUR Workshop Proceedings*, *3159*, 191–199. https://ceur-ws.org/Vol-3159/T1-19.pdf

Lee, M. (2023). Mathematical Analysis and Performance Evaluation of the GELU Activation Function in Deep Learning. *Journal of Mathematics*, *2023*, 1–13. https://doi.org/10.1155/2023/4229924

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv E-Prints*, arXiv:1907.11692. http://arxiv.org/abs/1907.11692

Masud, S., Khan, M. A., Goyal, V., Akhtar, M. S., & Chakraborty, T. (2024). Probing Critical Learning Dynamics of PLMs for Hate Speech Detection. *Findings of the Association for Computational Linguistics: EACL 2024*, 826–845. https://doi.org/10.18653/v1/2024.findings-eacl.55

Mitra, A., & Sankhala, P. (2021). Multilingual Hate Speech and Offensive Content Detection using Modified Cross-entropy Loss. *Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December 13-17, 2021, CEUR Workshop Proceedings*, *3159*, 349–356. https://ceur-ws.org/Vol-3159/T1-34.pdf

Mnassri, K., Farahbakhsh, R., Chalehchaleh, R., Rajapaksha, P., Jafari, A. R., Li, G., & Crespi, N. (2024). A survey on multi-lingual offensive language detection. *PeerJ Computer Science*, *10*, e1934. https://doi.org/10.7717/peerj-cs.1934

Modha, S., Mandl, T., Shahi, G. K., Madhu, H., Satapara, S., Ranasinghe, T., & Zampieri, M. (2021). Overview of the HASOC Subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages and Conversational Hate Speech. *Forum for Information Retrieval Evaluation*, 1–3. https://doi.org/10.1145/3503162.3503176

Ocampo, N., Sviridova, E., Cabrio, E., & Villata, S. (2023). An In-depth Analysis of Implicit and Subtle Hate Speech Messages. *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 1997–2013. https://doi.org/10.18653/v1/2023.eacl-main.147

Pérez, J. M., Luque, F. M., Zayat, D., Kondratzky, M., Moro, A., Serrati, P. S., Zajac, J., Miguel, P., Debandi, N., Gravano, A., & Cotik, V. (2023). Assessing the Impact of Contextual Information in Hate Speech Detection. *IEEE Access*, *11*, 30575–30590. https://doi.org/10.1109/ACCESS.2023.3258973

Roychowdhury, S., & Gupta, V. (2023). Data-Efficient Methods For Improving Hate Speech Detection. *Findings of the Association for Computational Linguistics: EACL 2023*, 125–132. https://doi.org/10.18653/v1/2023.findings-eacl.9

Salehin, I., & Kang, D.-K. (2023). A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain. *Electronics*, *12*(14), 3106. https://doi.org/10.3390/electronics12143106

Subramanian, M., Easwaramoorthy Sathiskumar, V., Deepalakshmi, G., Cho, J., & Manikandan, G. (2023). A survey on hate speech detection and sentiment analysis using machine learning and deep learning models. *Alexandria Engineering Journal*, *80*, 110–121. https://doi.org/10.1016/j.aej.2023.08.038

Szeghalmy, S., & Fazekas, A. (2023). A Comparative Study of the Use of Stratified Cross-Validation and Distribution-Balanced Stratified Cross-Validation in Imbalanced Learning. *Sensors*, *23*(4), 2333. https://doi.org/10.3390/s23042333

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. ukasz, & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, *30*. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Yadav, A., Khan, F. A., & Singh, V. (2024). A Multi-Architecture Approach for Offensive Language Identification Combining Classical Natural Language Processing and BERT-Variant Models. *Applied Sciences*, *14*(23), 11206. https://doi.org/10.3390/app142311206

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 885-898
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10512

Yin, W., & Zubiaga, A. (2021). Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science*, *7*, e598. https://doi.org/10.7717/peerj-cs.598

Zaghouani, W., Mubarak, H., & Biswas, Md. R. (2024). So Hateful! Building a Multi-Label Hate Speech Annotated Arabic Dataset. *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 15044–15055. https://aclanthology.org/2024.lrec-main.1308/