Klasifikasi Polaritas Meme Berbasis Deep Learning dan Clustering dengan Penerapan Deteksi Teks Sarkasme

1st Zeva Patu Assyadid
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
zevapt@student.telkomuniversity.ac.id

2nd Ema Rachmawati
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
emarachmawati@telkomuniveristy.ac.id

Abstrak*— Meme merupakan konten internet populer yang cepat menyebar di media sosial dan sering digunakan untuk mengekspresikan ide, kritik, atau ketertarikan. Namun, interpretasi meme dapat bervariasi sehingga menimbulkan tantangan dalam analisis sentimen, karena satu meme bisa dipandang positif atau negatif oleh individu berbeda. Untuk mengatasi hal tersebut, diperlukan sistem otomatis yang mampu memprediksi polaritas sentimen secara konsisten. Meme bersifat multimodal karena menggabungkan komponen visual dan teks, sehingga cocok untuk penelitian analisis sentimen berbasis multimodal. Penelitian ini mengusulkan model deep learning gabungan BERT dan DenseNet121, yang mengintegrasikan teks, gambar, serta fitur cluster berbasis face encoding. Untuk meningkatkan pemahaman konteks teks, BERT digunakan dengan pelatihan deteksi sarkasme. Dataset yang digunakan adalah SemEval 2020 Task 8: Memotion Analysis, yang menyediakan anotasi lengkap tentang sentimen dan sarkasme pada meme. Hasil penelitian menunjukkan bahwa model gabungan dengan deteksi sarkasme mencapai Macro-F1 sebesar 0.3047 dan akurasi 0.3738, melampaui baseline resmi (0.2176) dengan peningkatan sekitar 40%. Model ini juga lebih efektif dalam mendeteksi sentimen positif dan netral serta mengurangi kesalahan negatif palsu akibat sarkasme. Hal ini membuktikan bahwa integrasi deteksi sarkasme memperkuat performa klasifikasi sentimen pada meme.

***Kata kunci*— klasifikasi sentimen, meme, deep learning, clustering, deteksi sarkasme, memotion**

I. PENDAHULUAN

Dalam beberapa tahun terakhir, *meme* menjadi populer di berbagai *platform* media sosial sebagai sarana mengekspresikan perasaan atau berbagi lelucon [1]. Namun, perbedaan interpretasi dapat membuat sebagian orang merasa tidak nyaman, sehingga menimbulkan tantangan dalam analisis sentimen. Satu *meme* dapat dipandang positif atau negatif oleh individu berbeda, sehingga dibutuhkan sistem otomatis untuk memprediksi polaritas sentimen secara konsisten.

Polaritas sentimen merupakan bidang umum dalam pemrosesan bahasa alami. Dengan sifat *multimodal*, *meme* menjadi objek yang sesuai untuk analisis ini. Analisis *multimodal* menggabungkan berbagai *input* seperti gambar, teks, video, atau audio, sehingga mampu menangkap konteks lebih akurat dan sering memberikan hasil lebih baik dibandingkan unimodal [2]. Salah satu penelitian *multimodal* dengan performa teratas dalam kompetisi SemEval 2020: Memotion Analysis [3] adalah model Guoym [4], yang mengusulkan dua jenis *ensemble*, yaitu berbasis data dan berbasis fitur. Meski demikian, sebagian besar penelitian masih menggunakan model umum tanpa memanfaatkan fitur *multimodal* khusus pada meme, seperti fitur wajah, sehingga performanya belum optimal.

Model CDEL oleh Guo dkk. [5] memanfaatkan fitur wajah pada *meme* dan berhasil mencapai state-of-the-art, melampaui model sebelumnya. Prosesnya diawali dengan pemilihan fitur utama, yaitu gambar dan teks, kemudian dilanjutkan dengan pemilihan algoritma *clustering* serta klasifikasi untuk memperoleh performa gabungan terbaik. Tahap akhir mengintegrasikan hasil tersebut dengan model *deep learning*. Meskipun unggul dalam pemanfaatan fitur multimodal, CDEL masih lemah dalam mendeteksi sarkasme pada teks *meme*. Guo dkk. melaporkan tingkat kesalahan analisis sarkasme sebesar 38%, yang muncul ketika teks menyiratkan emosi berlawanan dengan makna sebenarnya.

GAMBAR 1
Contoh Eror Analisis Deteksi Teks Sarkasme

Seperti ditunjukkan pada Gambar 1, salah satu eror deteksi sarkasme pada model CDEL terjadi pada teks "*launch the nukes*" yang secara eksplisit bermakna negatif, namun sebenarnya berlabel positif. Deteksi sarkasme merupakan sub-tugas menantang dalam analisis sentimen karena membutuhkan pemahaman makna tersirat yang sering berlawanan dengan makna literal [6]. Sarkasme juga dipengaruhi konteks, nada, dan referensi kultural yang tidak selalu tersurat [7]. Model konvensional sering salah

mengklasifikasikan teks sarkastik karena hanya melihat semantik permukaan, sedangkan BERT mampu menangkap kontras halus dalam sentimen melalui mekanisme bidirectional attention, sehingga lebih efektif dalam pendeteksian sarkasme [6][8].

Penelitian ini mengembangkan sistem analisis polaritas sentimen *meme* dengan memanfaatkan fitur wajah sebagai fitur berbasis *cluster*, terinspirasi dari model CDEL, namun dilengkapi deteksi teks sarkasme pada model teks. Model teks dilatih menggunakan dataset sarkasme guna meningkatkan pemahaman konteks sarkastik dalam meme. Fokus penelitian ini adalah menekankan pentingnya integrasi deteksi sarkasme dalam model *deep learning* gabungan untuk meningkatkan performa analisis sentimen *meme*.

II. KAJIAN TEORI

A. DenseNet

DenseNet (*Densely Connected Convolutional Networks*) diperkenalkan oleh Gao Huang dkk. pada 2017 [10], dengan ciri utama *dense connectivity* antar *layer*. Setiap *layer* menerima *input* dari semua *layer* sebelumnya dan meneruskan hasil ekstraksinya melalui konkatenasi, bukan penjumlahan. Mekanisme ini memungkinkan pemanfaatan ulang fitur secara efisien dan memperdalam representasi. Pada penelitian ini, DenseNet digunakan karena kemampuannya mengekstraksi fitur kompleks dari gambar meme [11], yang sering memuat elemen visual beragam seperti teks, ekspresi wajah, simbol, maupun konteks visual tertentu.

Arsitektur DenseNet terdiri dari beberapa komponen utama: (1) *Dense Block*, berisi beberapa *layer* konvolusi dengan *output* setiap *layer* yang dikonkatenasi ke *input layer* berikutnya; (2) *Transition Layer*, yang menghubungkan dense block menggunakan 1×1 *convolution* dan *average pooling* untuk *downsampling*; (3) *Pooling Layer*, yang mereduksi *output* menjadi vektor fitur; dan (4) *Classification Layer,* berupa *fully connected layer* dan *softmax*.

Model yang digunakan adalah DenseNet121, varian dengan 121 *layer* dan empat *dense block*. *Pretrained weights* dari ImageNet dimanfaatkan untuk *transfer learning*, dengan beberapa *layer* akhir yang dapat di-*fine-tune* agar sesuai dengan karakteristik visual *meme* dalam *dataset* [10].

B. BERT

BERT (*Bidirectional Encoder Representations from Transformers*) adalah model bahasa berbasis *Transformer* yang diperkenalkan oleh Devlin dkk. pada 2018 [12]. Berbeda dari pendekatan konvensional yang hanya memproses teks satu arah, BERT dirancang memahami konteks kata secara dua arah (*bidirectional*), sehingga mampu menangkap makna mendalam dan relasi semantik kompleks dalam kalimat.

Input teks ke BERT melalui tahap tokenisasi dengan WordPiece *tokenizer*, yang memecah kata menjadi sub-kata agar dapat menangani *out-of-vocabulary*. Token hasil pemrosesan dikonversi menjadi *input ID, token type ID*, dan *attention mask*, kemudian diproses dalam lapisan BERT. Representasi keseluruhan teks biasanya diambil dari token khusus [CLS].

Arsitektur BERT-base terdiri atas 12 *encoder layer*, 12 *attention head*, dan ukuran *hidden layer* 768. Setiap *encoder layer* mencakup dua komponen utama: (1*) multi-head self-attention*, yang memungkinkan model memperhatikan hubungan antar kata dari berbagai konteks secara paralel, dan (2) *feed-forward neural network*, yang memperkuat hasil atensi.

C. FaceNet

FaceNet adalah model *deep learning* berbasis *Convolutional Neural Network* (CNN) yang menghasilkan representasi vektor wajah dalam bentuk *embedding* berdimensi tetap. Model ini diperkenalkan oleh Schroff dkk. pada 2015 [13] dengan tujuan utama pengenalan, verifikasi, dan *clustering* wajah. Alih-alih melakukan klasifikasi langsung, FaceNet mengubah citra wajah menjadi vektor numerik yang merepresentasikan identitas di ruang vektor.

Schroff dkk. menyebut FaceNet terinspirasi dari arsitektur Inception-ResNet v1 milik David Sandberg [14], sehingga model ini memanfaatkan *pretrained weights* milik David. Detail komponen dapat ditemukan lebih lanjut dalam dokumentasi FaceNet [13].

Dalam penelitian ini, FaceNet digunakan untuk mengekstraksi fitur wajah dari gambar *meme* yang kemudian diproses melalui *clustering*. FaceNet dipilih karena mampu menghasilkan *embedding* stabil yang dapat dibandingkan langsung menggunakan metrik jarak, seperti *Euclidean distance,* sehingga sesuai untuk *metode hierarchical clustering.*

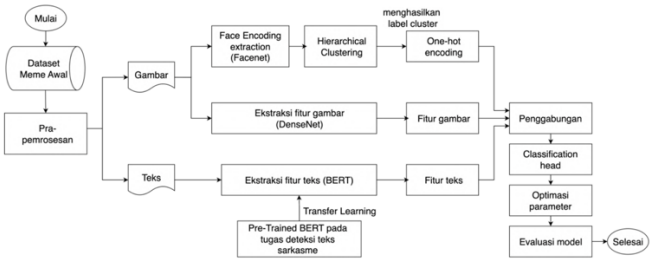
D. Hierarchical Clustering

Hierarchical clustering adalah metode pengelompokan data berdasarkan tingkat kemiripan tanpa memerlukan label. Terdapat dua pendekatan, yaitu *divisive* (pemisahan) dan *agglomerative* (penggabungan). Penelitian ini menggunakan *agglomerative clustering* [15], di mana setiap data awalnya dianggap sebagai satu *cluster*, lalu digabungkan bertahap dengan *cluster* lain berdasarkan ukuran kedekatan hingga terbentuk hierarki penuh. Sebagai ukuran kedekatan digunakan *Euclidean distance*, yang efektif untuk data numerik berdimensi tetap seperti vektor wajah. Proses penggabungan *cluster* menerapkan *Ward linkage*, yang meminimalkan total variansi dalam *cluster* sehingga menghasilkan kelompok yang lebih homogen.

III. METODE

A. Kerangka Kerja

Model *deep learning* gabungan yang diusulkan dirancang dengan kerangka kerja seperti ditunjukkan pada Gambar 2.



GAMBAR 2
Contoh Kerangka Kerja Model yang Dibangun

B. Dataset

Dataset yang digunakan dalam penelitian ini adalah *Dataset* Memotion dari kompetisi SemEval 2020 Task 8 [3].

Dataset ini telah dibagi oleh penyelenggara menjadi 6992 data latih dan 1878 data uji. Distribusi kelas sentimen pada kedua *subset* ditunjukkan pada Tabel 1, dengan rincian:

TABEL 1
DISTRIBUSI SENTIMEN PADA DATA LATIH DAN UJI

	Data Latih	Data Uji
Positif	4160	1111
Netral	2201	594
Negatif	631	173

Dataset Memotion memiliki total 8 kolom, yaitu *image_name, text_ocr, text_corrected, humour, sarcasm, offensive,* dan *motivational*. Namun, penelitian ini hanya memanfaatkan kolom nama gambar, teks, label sarkasme, serta label sentimen.

C. *Preprocess*

Preprocess yang dilakukan mencakup 3 tahap utama, yaitu sebagai berikut:

a. Pemrosesan Teks Awal

Pertama, untuk mengatasi nilai *null* pada kolom teks, dilakukan strategi sebagai berikut: apabila kolom *text_corrected* bernilai *null* namun kolom *text_ocr* memiliki nilai, maka isi dari *text_ocr* disalin ke *text_corrected*. Selanjutnya, sisa nilai *null* pada kolom *text_ocr* diganti dengan *string* kosong ("") dan seluruh kolom teks dikonversi ke tipe *string* agar proses tokenisasi dapat berjalan dengan benar.

b. Penyelarasan Label Sentimen

Pada data latih, label sentimen diseragamkan dengan menyatukan kategori serupa. Label *very_positive* diubah menjadi *positive*, sementara label *very_negative* diubah menjadi *negative*. Hal ini dilakukan untuk menjaga konsistensi kelas dengan label *ground truth* yang terdapat pada data uji.

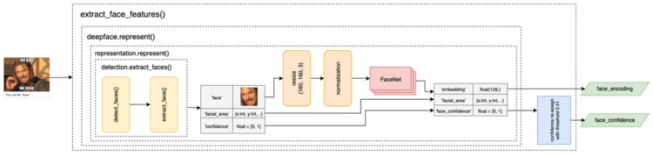
c. Pemrosesan Gambar

Gambar diproses menggunakan fungsi kustom *load_and_preprocess_image*, yang mencakup: (a) mengubah ukuran menjadi 224 × 224 piksel, (b) mengonversinya ke bentuk *array* NumPy, dan (c) menerapkan fungsi *preprocess_input* dari *library* keras.applications.resnet50 agar sesuai dengan kebutuhan ekstraksi fitur. Selama tahap ini, ditemukan sebuah IOError pada gambar image_5119.png yang teridentifikasi sebagai *corrupted image*. Masalah ini diatasi dengan mengatur parameter LOAD_TRUNCATED_IMAGES = True.

D. Implementasi Sistem

a. Ekstraksi Face Encoding

Untuk mengekstraksi *face encoding* dari gambar, penelitian ini memanfaatkan fungsi *represent* dari *library* DeepFace [16] dengan model Facenet. Proses ini diimplementasikan dalam sebuah fungsi kustom *extract_face_features*() seperti ditunjukkan pada Gambar 3. Fungsi tersebut menghasilkan representasi vektor dari wajah yang terdeteksi pada gambar. *Output* utama yang diperoleh berupa *embedding* (*face encoding*), yaitu representasi vektor numerik dari wajah, serta *face_confidence* yang merepresentasikan tingkat keyakinan terhadap hasil deteksi wajah.



GAMBAR 3
Skema Ekstraksi Face Encoding

b. Clustering Wajah

Setelah ekstraksi, dilakukan *agglomerative hierarchical clustering* dengan jarak *Euclidean* dan *linkage Ward* untuk mengelompokkan *meme* berdasarkan kesamaan *face encoding*. Penentuan jumlah *cluster* optimal dilakukan dengan mengevaluasi beberapa nilai *threshold* menggunakan *Silhouette Score, Calinski–Harabasz Score*, dan *Davies–Bouldin Index*, sehingga diperoleh *threshold* 14.9 dengan 614 *cluster*. Gambar tanpa wajah terdeteksi atau dengan *face confidence* di bawah 0.51 dimasukkan ke dalam *default cluster* (-1). *Centroid* dihitung untuk tiap *cluster* (kecuali *default*) dan digunakan pada tahap pengujian untuk memberikan label *cluster* dengan membandingkan jarak *Euclidean face encoding* data uji terhadap *centroid* terdekat.

c. Pretrain BERT pada Deteksi Teks Sarkasme

Model deteksi sarkasme dibangun di atas arsitektur 'bert-base-cased' dengan penambahan beberapa lapisan *Dense* beraktivasi *ReLU* dan *Dropout* untuk mencegah overfitting. Model ini dilatih untuk mengklasifikasikan teks ke dalam empat kategori sarkasme: *not_sarcastic, general, twisted_meaning,* dan *very_twisted*, menggunakan label pada kolom '*sarcasm*'. Setelah pelatihan, bobot model disimpan dan dimuat kembali pada arsitektur yang sama untuk merekonstruksi model tiruan. Dari model tiruan tersebut, komponen BERT hasil *fine-tuning* diekstraksi dan digunakan sebagai lapisan BERT dalam model utama *multimodal*, sehingga memungkinkan *transfer* representasi yang lebih kaya dari deteksi sarkasme ke tahap klasifikasi sentimen.

d. Text Encoding

Penelitian ini memanfaatkan model *pretrained* BERT 'bert-base-cased' dengan *transfer learning* dari korpus bahasa Inggris berskala besar. Proses *encoding* dilakukan melalui tokenisasi WordPiece, menghasilkan *input_ids* dan *token_type_ids* sebagai masukan ke BERT. Model menghasilkan *embedding* kontekstual untuk setiap token, dari mana representasi teks diperoleh melalui token [CLS] serta *mean* dan *max pooling* pada *embedding* token. Fitur yang dihasilkan kemudian dikonkatenasi dan diproses lebih lanjut menggunakan beberapa lapisan *Dense, Batch Normalization*, dan *Dropout* untuk reduksi dimensi dan ekstraksi representasi optimal.

e. Image Encoding

Gambar diekstraksi menggunakan DenseNet121 dengan bobot *pretrained* ImageNet, di mana lapisan akhir dibuat trainable untuk *fine-tuning* pada *dataset* Memotion. Ekstraksi dilanjutkan dengan *global average pooling* dan *global max pooling* untuk mereduksi dimensi dan menangkap fitur representatif. Vektor hasil *pooling* digabung, lalu diproses

melalui *dense layer, batch normalization*, dan *dropout* guna meningkatkan stabilitas pembelajaran serta mencegah *overfitting*.

f. One-Hot Encoding Fitur Cluster

Setelah setiap gambar *meme* memperoleh label *cluster*, label tersebut diubah menjadi *format one-hot encoding* sebagai fitur *input* pada model gabungan. *One-hot encoding* merepresentasikan setiap *cluster* dalam bentuk vektor biner berdimensi k (jumlah total *cluster*), dengan nilai 1 pada posisi label terkait dan 0 pada posisi lainnya. Representasi ini memungkinkan informasi kategorikal dari *cluster* diproses secara efektif oleh lapisan *dense*.

g. Klasifikasi

Ketiga fitur digabungkan melalui lapisan konkatenasi menjadi satu vektor fitur komprehensif yang kemudian digunakan sebagai input untuk *classification head*. Klasifikasi dilakukan dengan serangkaian *lapisan Dense (fully connected)* yang dilengkapi *batch normalization* dan *dropout* untuk meningkatkan stabilitas serta mencegah *overfitting*. Lapisan *Dense* terakhir memiliki tiga unit *output* yang merepresentasikan kelas target (-1, 0, 1). Fungsi aktivasi *softmax* diterapkan untuk menghasilkan distribusi probabilitas pada tiap kelas, dengan prediksi ditentukan berdasarkan probabilitas tertinggi.

h. Focal Loss

Untuk menangani ketidakseimbangan kelas pada *dataset* Memotion, penelitian ini menerapkan *Focal Loss* [17], yang memodifikasi *categorical cross-entropy* dengan faktor modulasi $(1-p_t)^{\gamma}$ agar model lebih fokus pada sampel minoritas atau sulit diklasifikasikan. *Focal Loss* ($FL(p_t)$) didefinisikan sebagai:

$$FL(p_t) = -\alpha_t(1-\mathrm{p}_t)^{\gamma}\log(\mathrm{p}_t) \quad (1)$$

Dengan p_t merupakan probabilitas prediksi kelas sebenarnya, α_t bobot kelas, dan γ parameter fokus. Dalam penelitian ini, γ ditetapkan sebesar 2.0 dan bobot kelas α_t diatur menjadi [1.0,2.0,3.0][1.0, 2.0, 3.0][1.0,2.0,3.0] untuk kelas negatif, netral, dan positif secara berurutan, sehingga mengurangi dominasi kelas mayoritas serta meningkatkan kontribusi sampel minoritas.

i. Callback Pelatihan

Penelitian ini mengoptimalkan proses pelatihan dengan memanfaatkan beberapa *callback*, yaitu *ModelCheckpoint, EarlyStopping,* dan *ReduceLROnPlateau*. *ModelCheckpoint* digunakan untuk menyimpan bobot terbaik ketika terjadi peningkatan Macro-F1 pada data validasi, sementara *EarlyStopping* menghentikan pelatihan jika tidak ada peningkatan Macro-F1 selama 10 epoch berturut-turut guna mencegah *overfitting*. Selain itu, *ReduceLROnPlateau* menyesuaikan nilai *learning rate* secara dinamis dengan menurunkannya sebesar faktor 0,75 ketika kinerja validasi stagnan selama 5 epoch, dengan batas minimum 1E-6, sehingga model tetap dapat beradaptasi dalam proses pelatihan.

E. Evaluasi

a. Evaluasi Performa Clustering

Metode evaluasi *clustering* terbagi menjadi indikator eksternal dan internal [18]. Karena dataset tidak memiliki label asli, penelitian ini menggunakan indikator internal, yaitu *Silhouette Score* (SC), *Calinski-Harabasz Score* (CHS), dan *Davies-Bouldin Index* (DBI). SC menilai konsistensi sampel dalam *cluster*, CHS mengukur rasio variansi antar–intra *cluster*, sedangkan DBI menilai keterpisahan antar *cluster*. Kombinasi ketiganya memberikan evaluasi yang seimbang antara koherensi internal dan separasi *cluster*. Untuk metrik komprehensif, penelitian ini mengadopsi *Comprehensive Indicator* (CI) sesuai definisi Guo dkk. [5].

$$CI = SC_N + CHS_N - DBI_N \quad (2)$$

Adapun SC_N, CHS_N, DBI_N adalah metrik SC, CHS, dan DBI yang dinormalisasi.

Metrik SC_N, CHS_N, dan DBI_N didapatkan dengan perhitungan sebagai berikut:

$$SC_N = \frac{SC - SC_{Min}}{SC_{Max} - SC_{Min}} \quad (3)$$

$$CHS_N = \frac{CHS - CHS_{Min}}{CHS_{Max} - CHS_{Min}} \quad (4)$$

$$DBI_N = 1 - \frac{DBI - DBI_{Min}}{DBI_{Max} - DBI} \quad (5)$$

Metrik-metrik ini dihitung dengan berbagai *threshold* yang diterapkan pada *cluster face encoding*. *Threshold* (t) optimal ditentukan dengan memilih nilai yang menghasilkan CI maksimum.

b. Evaluasi Performa Klasifikasi

Berdasarkan evaluasi resmi dari penyedia *dataset* Memotion, penelitian ini menggunakan Macro-F1 score sebagai metrik utama. Metrik ini sesuai untuk *dataset* tidak seimbang karena menghitung *F1 score* pada setiap kelas secara terpisah, lalu merata-ratakannya sehingga setiap kelas memiliki bobot yang sama. Untuk memberikan gambaran yang lebih menyeluruh, penelitian ini juga menghitung rata-rata tertimbang (*weighted average*) dari *Precision, Recall,* dan *F1 score*, yang mempertimbangkan proporsi setiap kelas dalam dataset. Selain itu, evaluasi per kelas dilakukan dengan menilai *Precision, Recall,* dan *F1 score* pada masing-masing kelas sentimen. Terakhir, *confusion matrix* divisualisasikan untuk lebih memahami performa model di seluruh kelas sentimen.

F. Percobaan

Bagian percobaan dalam penelitian ini dirancang untuk menyoroti perbandingan performa antara model dengan BERT dasar dan model dengan BERT yang telah dilatih untuk deteksi teks sarkasme. Tiga skenario utama digunakan dalam eksperimen ini.

Pertama, pada Skenario 1: Rasio *Validation Split*, dilakukan variasi rasio *split* sebesar 0.10, 0.15, 0.20, dan 0.25 untuk menilai *robustness* serta konsistensi performa model. Variasi ini memungkinkan pengamatan terhadap bagaimana

perbedaan proporsi data pelatihan dan validasi memengaruhi kemampuan generalisasi model dalam klasifikasi.

Kedua, pada Skenario 2: Pengaturan *Learning Rate*, dilakukan percobaan dengan nilai 5E-6, 1E-5, 2E-5, dan 5E-5. Tujuannya adalah untuk mengevaluasi pengaruh pengaturan *learning rate* terhadap performa model secara keseluruhan serta menemukan nilai optimal untuk stabilitas pelatihan.

Ketiga, pada Skenario 3: Strategi Bobot *Kelas*, digunakan tiga pendekatan: bobot default, bobot *balanced* yang dihitung secara komputasi, dan bobot berdasarkan rasio distribusi kelas. Strategi *balanced* memberikan bobot lebih tinggi pada kelas minoritas melalui perhitungan terbalik terhadap frekuensi, sedangkan bobot berbasis distribusi kelas menggunakan proporsi relatif tiap kelas pada dataset.

Dengan kombinasi ketiga skenario tersebut, eksperimen ini bertujuan untuk memberikan gambaran komprehensif mengenai faktor-faktor yang memengaruhi performa model dalam klasifikasi sentimen multimodal dengan deteksi sarkasme.

IV. HASIL DAN PEMBAHASAN

A. Analisis Hasil Percobaan

Percobaan dilakukan pada data uji dengan menjalankan model sampai *epoch* terhenti. Hasil percobaan Skenario 1 pada Tabel 2 menunjukkan bahwa rasio *validation split* berpengaruh signifikan terhadap performa model. Nilai Macro-F1 terbaik diperoleh pada *split* 0,25, yaitu 0,3047 untuk model sarkasme dan 0,3095 untuk model dasar. Alokasi validasi yang lebih besar (25%) membuat proses evaluasi selama pelatihan lebih andal, sehingga meningkatkan kemampuan generalisasi model. Sebaliknya, rasio yang lebih kecil seperti 0,15 menghasilkan performa lebih rendah akibat keterbatasan data validasi dan ketidakstabilan dalam pembelajaran.

TABEL 2
HASIL MACRO-F1 DARI SKENARIO 1: RASIO VALIDATION SPLIT

Rasio Validation Split	Sarkasme	Dasar
0.1	0.2786	0.2460
0.15	0.2198	0.2226
0.2	0.2821	0.2479
0.25	**0.3047**	**0.3095**

TABEL 3
HASIL MACRO-F1 DARI SKENARIO 2: PENGATURAN LEARNING RATE

Learning Rate	Sarkasme	Dasar
1E-6	0.2082	0.1289
5E-6	**0.3047**	**0.3095**
1E-5	0.2697	0.2750
2E-5	0.1978	0.2107

Hasil pada Tabel 3 selanjutnya menyoroti pengaruh penggunaan *batch size* terhadap performa model. Nilai Macro-F1 terbaik diperoleh pada *batch size* 16 dengan capaian 0,3095 untuk model dasar dan 0,3047 untuk model sarkasme. Pada *batch size* yang lebih kecil seperti 8, performa model menurun karena gradien yang dihasilkan kurang stabil sehingga menghambat proses konvergensi. Sebaliknya, *batch size* yang lebih besar seperti 32 juga menurunkan performa akibat berkurangnya kemampuan model dalam menangkap variasi gradien, yang menyebabkan proses generalisasi menjadi kurang optimal. Hal ini menunjukkan bahwa pemilihan *batch size* yang seimbang berperan penting dalam menjaga stabilitas pembelajaran sekaligus mempertahankan kemampuan generalisasi model.

TABEL 4
HASIL MACRO-F1 DARI SKENARIO 3: STRATEGI BOBOT KELAS

Bobot Kelas	Sarkasme	Dasar
Default (1)	**0.3047**	**0.3095**
Balanced	0.2331	0.2454
Distribusi kelas	0.2237	0.2778

Hasil pada Tabel 4 memperlihatkan bahwa variasi strategi pembobotan kelas berdampak nyata terhadap performa model. Baik pada model sarkasme maupun model dasar, pengaturan bobot default (1) justru memberikan nilai Macro-F1 tertinggi, masing-masing 0,3047 dan 0,3095. Sebaliknya, penggunaan bobot *balanced* menurunkan performa secara signifikan, sedangkan pendekatan distribusi kelas menghasilkan perbedaan efek: model dasar sedikit meningkat dibandingkan *balanced* (0,2778 vs 0,2454), sementara model sarkasme justru semakin melemah. Temuan ini menegaskan bahwa dalam skenario klasifikasi sentimen *multimodal*, strategi pembobotan ulang sering kali memunculkan ketidakstabilan dan potensi kompensasi berlebihan terhadap kelas minoritas, sehingga bobot default cenderung menghadirkan hasil yang lebih stabil dan efektif.

B. Analisis Hasil Pengujian

Seperti terlihat pada hasil percobaan, model dasar memperoleh Macro-F1 sedikit lebih tinggi (0,3095) dibanding model sarkasme (0,3047), dengan akurasi masing-masing 0,3488 dan 0,3738. Untuk memberikan gambaran lebih menyeluruh, tabel berikut turut menyajikan metrik evaluasi tambahan, meliputi metrik tertimbang serta performa per kelas dari kedua model.

TABEL 5
PERBANDINGAN METRIK EVALUASI KESELURUHAN

Metrik	Sarkasme	Dasar
Macro-F1	0.3047	**0.3095**
Akurasi	**0.3778**	0.3488
F1 Score Tertimbang	**0.3735**	0.3631
Precision Tertimbang	**0.4588**	0.4531
Recall Tertimbang	**0.3738**	0.3488

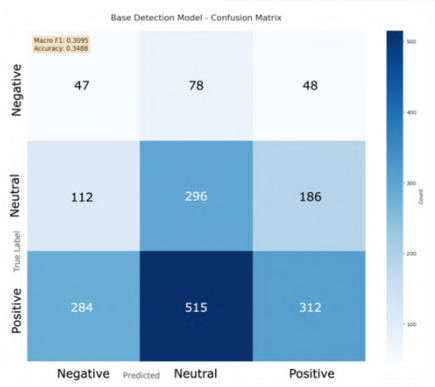
Berdasarkan Tabel 6, performa tiap kelas memperlihatkan keunggulan yang berbeda antara model sarkasme dan model dasar. Pada kelas negatif, model dasar lebih unggul pada metrik recall (0,2717 dibanding 0,1156) serta *F1 score* (0,1526 dibanding 0,1108), meskipun keduanya memiliki precision yang hampir sama. Temuan ini mengindikasikan bahwa model dasar lebih efektif dalam mengenali sampel negatif, walaupun tingkat precision yang rendah menegaskan masih adanya kesulitan dalam mendeteksi ekspresi negatif pada data meme.

TABEL 6
PERBANDINGAN METRIK EVALUASI KESELURUHAN

Kelas	Metrik	Sarkasme	Dasar
Positif	Precision	**0.1064**	0.1061
	Recall	0.1156	**0.2717**
	F1 Score	0.1108	**0.1526**
Netral	Precision	0.3101	**0.3330**
	Recall	**0.5909**	0.4983
	F1 Score	**0.4067**	0.3992
Negatif	Precision	**0.5932**	0.5714
	Recall	**0.2979**	0.2808
	F1 Score	**0.3966**	0.3766

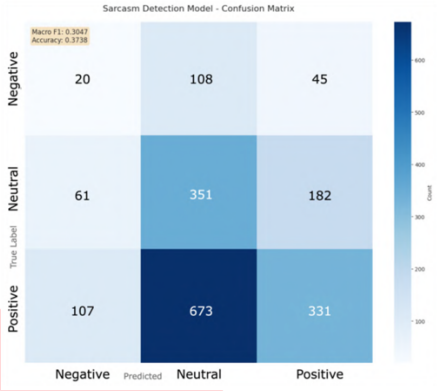
Sebaliknya, pada kelas netral, model sarkasme memperlihatkan kinerja yang lebih baik pada metrik recall (0,5909 dibanding 0,4983) dan *F1 score* (0,4067 dibanding 0,3992), meskipun model dasar memiliki keunggulan tipis pada *Precision* (0,3330 dibanding 0,3101). Hal ini menunjukkan bahwa model sarkasme lebih peka dalam mengenali sampel netral dan mampu mengklasifikasikannya dengan benar dalam jumlah yang lebih banyak, walaupun tingkat ketepatannya sedikit menurun dibandingkan model dasar.

Pada kelas positif, model sarkasme menunjukkan keunggulan pada seluruh metrik evaluasi, yaitu *Precision* (0,5932 dibanding 0,5714), recall (0,2979 dibanding 0,2808), serta *F1 score* (0,3966 dibanding 0,3766). Hasil ini menegaskan bahwa integrasi deteksi sarkasme mampu meningkatkan efektivitas model dalam mengenali sentimen positif, baik dari sisi ketepatan maupun jumlah sampel yang terklasifikasi dengan benar. Secara keseluruhan, temuan ini memperlihatkan bahwa penerapan deteksi sarkasme dapat memberikan perbaikan performa pada kelas tertentu, khususnya sentimen positif. Untuk memperoleh pemahaman yang lebih menyeluruh terkait kesalahan prediksi yang masih terjadi, analisis selanjutnya difokuskan pada evaluasi *confusion matrix* dari kedua model.



GAMBAR 4
Confusion Matrix Model Klasifikasi dengan BERT Dasar

Seperti ditunjukkan pada Gambar 4 dan Gambar 5, model dengan deteksi sarkasme mampu secara signifikan menurunkan jumlah *false negative*, khususnya pada kelas positif. Model ini hanya salah mengklasifikasikan 107 sampel positif sebagai negatif, jauh lebih sedikit dibandingkan 284 sampel pada model dasar. Peningkatan ini mencerminkan sensitivitas yang lebih baik dalam mengenali sentimen positif ketika elemen sarkasme diperhitungkan. Temuan tersebut selaras dengan hasil *F1 score* per kelas, di mana model sarkasme mencatat nilai lebih tinggi untuk kelas positif (0,3966 dibanding 0,3766), meskipun secara agregat Macro-F1 tidak mengalami peningkatan yang berarti.



GAMBAR 5
Confusion Matrix Model Klasifikasi dengan BERT Sarkasme

Selain itu, model dengan deteksi sarkasme memperlihatkan keseimbangan yang lebih baik dalam performa antar kelas. Model ini memperoleh *F1 score* tertinggi pada kelas netral (0,4067) serta menunjukkan peningkatan performa pada kelas positif (0,3966 dibandingkan 0,3766 pada model dasar). Meskipun pada kelas negatif model dasar masih sedikit lebih unggul (0,1526 dibandingkan 0,1108), model sarkasme tetap menonjol melalui distribusi precision dan recall yang lebih seimbang di ketiga kelas, sehingga mengurangi dominasi satu kelas tertentu dan memperlihatkan kemampuan klasifikasi yang lebih konsisten.

Analisis ini mengindikasikan bahwa meskipun model dasar sedikit lebih unggul pada Macro-F1 secara keseluruhan, model sarkasme menunjukkan ketahanan yang lebih baik, khususnya dalam mengenali sentimen positif dan menekan *false negative* pada konteks sarkastik. *Trade-off* ini menegaskan potensi model sarkasme dalam meningkatkan sensitivitas terhadap nuansa ekspresi pada data yang mengandung unsur sarkasme.

C. Analisis Sampel Eror

Untuk memahami lebih jauh keterbatasan model sarkasme, dianalisis sejumlah sampel yang salah diklasifikasikan pada tiap kelas.

a. Sampel pada Kelas Positif

Pada kelas positif, model sarkasme masih menghasilkan jumlah kesalahan yang cukup besar, yaitu 673 sampel diklasifikasikan sebagai netral dan 107 sebagai negatif. Analisis terhadap *confidence score* menunjukkan bahwa kesalahan ke kelas netral umumnya disertai perbedaan (*delta*) *confidence* yang relatif kecil antara kelas positif dan netral, dengan rata-rata 0.3002 dan standar deviasi 0.1942. Hal ini mengindikasikan bahwa model sering kurang yakin dalam membedakan keduanya, sehingga rentan menimbulkan ambiguitas.

Sampel pada Gambar 6 menampilkan seorang anak dengan telinga menonjol dan pria dewasa memakai headphone yang seharusnya bernuansa positif, namun diprediksi model sebagai negatif. Kesalahan ini muncul karena model gagal menangkap konteks humor visual serta

narasi positif tentang transformasi dan penerimaan diri. Sebaliknya, model memberi bobot besar pada teks *'Let me see your ears'* yang ditafsirkan sebagai ejekan, serta elemen visual telinga menonjol sebagai konotasi negatif. Hal ini tercermin dari skor *confidence*, di mana kelas negatif memperoleh 0.4066, positif 0.4037, dan netral 0.1897. Selisih yang sangat kecil antara kelas negatif dan positif ($\Delta \approx 0.0029$) menunjukkan bahwa model sebenarnya hampir memilih kelas benar, namun condong ke negatif akibat interpretasi harfiah terhadap teks dan citra.

GAMBAR 6
Sampel Eror Klasifikasi Sentimen Positif ke Negatif

b. Sampel pada Kelas Netral

Dari distribusi *delta confidence*, terlihat bahwa kesalahan klasifikasi pada kelas netral lebih sering bergeser ke arah positif dibandingkan negatif, dengan jumlah kasus masing-masing 182 dan 61. Pola ini mengisyaratkan adanya kecenderungan bias pada model untuk menafsirkan konten netral sebagai bernuansa positif. Selain itu, rata-rata *delta confidence* pada kesalahan ke positif (0.212) sedikit lebih tinggi dibandingkan pada kesalahan ke negatif (0.197). Artinya, dalam kondisi keliru tersebut, keyakinan model terhadap prediksi positif umumnya lebih kuat dibandingkan terhadap negatif.

GAMBAR 7
Sampel Eror Klasifikasi Sentimen Netral ke Positif

Gambar 7 menampilkan sampel *meme* netral yang diprediksi positif oleh model, menggambarkan Spiderman bergelantungan dengan teks "*Mommy come quick! I think I saw a spider*". Humor ironi muncul dari kontradiksi antara Spiderman yang berkuasa atas laba-laba namun justru takut pada laba-laba. Meskipun teks secara literal netral, model menafsirkan sentimen positif karena karakter Spiderman membawa konotasi positif yang kuat dan humor yang tercipta memperkuat kesan menghibur. Hal ini menunjukkan bahwa model cenderung memprioritaskan elemen visual dan konteks humor yang menyiratkan sentimen positif, sehingga pergeseran dari netral ke positif bisa dianggap wajar.

c. Sampel pada Kelas Negatif

Analisis kesalahan menunjukkan bahwa prediksi negatif yang salah lebih sering diklasifikasikan sebagai netral daripada positif, mengindikasikan kesulitan model dalam mendeteksi sentimen negatif yang tersirat dalam *meme*. Rata-rata *delta confidence* untuk kesalahan negatif-ke-netral (0,358) lebih tinggi dibandingkan kesalahan negatif-ke-positif (0,283), menunjukkan bahwa model cenderung lebih yakin saat salah menafsirkan *meme* negatif sebagai netral.

GAMBAR 6
Sampel Eror Klasifikasi Sentimen Negatif ke Netral

Pada sampel dalam Gambar 8, model salah memprediksi *meme* sarkastik yang berlabel negatif sebagai netral. *Meme* menampilkan seorang laki-laki di depan papan tulis dengan teks "*IF YOUR FRIENDS DID'NT WISH YOU HAPPY FRIENDSHIP DAY, YOU HAVE AMAZING FRIENDS*" di mana makna negatif tersirat melalui sindiran sarkastik: *punchline* "*you have amazing friends*" sebenarnya menekankan penolakan sosial dan kesepian. Model gagal menangkap ironi ini karena teks mengandung kata-kata positif yang kuat seperti *"friend", "friendship", "happy",* dan *"amazing"* sehingga sinyal positif mendominasi penilaian model. Dalam kondisi adanya kontradiksi antara kata-kata positif dan konteks negatif, model cenderung menetapkan kelas netral sebagai prediksi dengan confidence tinggi (0,762), dibandingkan confidence untuk kelas negatif (0,115) dan positif (0,123). Hal ini menunjukkan bahwa model lebih memilih "bermain aman" ketika menghadapi ambiguitas atau sinyal yang saling bertentangan, menandakan keterbatasannya dalam memahami sarkasme dan nuansa sentimen yang tersembunyi.

V. KESIMPULAN

Secara keseluruhan, penelitian ini berhasil mengembangkan sistem klasifikasi polaritas sentimen dengan *dataset* Memotion menggunakan arsitektur *multimodal*. Model memadukan informasi tekstual dengan deteksi sarkasme melalui BERT, konten visual menggunakan DenseNet121, serta fitur *cluster* wajah untuk menangkap nuansa dan emosi tersirat dalam meme. Hasil pengujian menunjukkan bahwa model sarkasme mencapai Macro-F1 sebesar 0,3047 dengan akurasi 0,3738, sedangkan model dasar sedikit lebih tinggi pada Macro-F1 (0,3095). Meski demikian, model sarkasme lebih unggul dalam mendeteksi sentimen positif dan netral serta mampu mengurangi *false negative*, menunjukkan kemampuan yang lebih baik dalam menangkap konteks sarkastik dan nuansa emosi yang kompleks.

REFERENSI

[1] S. Tabatabaei dan E. A. Ivanova, "The role of memes on emotional contagion", Elementary Education Online, vol. 20, no. 5, hlm. 6028–6036, 2023. [Daring]. Tersedia: https://ilkogretim-online.org/index.php/pub/article/view/4690

[2] S. Kholmatov, "Multimodal sentiment analysis: A study on emotion understanding and classification by integrating text and images," Beijing Institute of

Technology, 2023. [Daring]. Tersedia: https://doi.org/10.5281/zenodo.13909963
[3] C. Sharma, D. Bhageria, W. Scott et al., “SemEval-2020 Task 8: Memotion analysis – The visuo-lingual metaphor!” dalam Proceedings of the 12th International Semantic Evaluation Conference (SemEval-2020), 2020. [Daring]. Tersedia: https://aclanthology.org/2020.semeval-1.99.pdf
[4] Y. Guo, J. Huang, Y. Dong et al., “Guoym at SemEval-2020 Task 8: Ensemble-based classification of visuo-lingual metaphor in memes,” dalam Proceedings of the Fourteenth Workshop on Semantic Evaluation, Barcelona (online): International Committee for Computational Linguistics, 2020, hlm. 1120–1125. [Daring]. Tersedia: https://aclanthology.org/2020.semeval-1.148
[5] X. Guo, J. Ma, dan A. Zubiaga, “Cluster-based deep ensemble learning for emotion classification in internet memes,” arXiv preprint arXiv:2302.08343, 2023. [Daring]. Tersedia: https://doi.org/10.48550/arXiv.2302.08343
[6] W. Chen, F. Lin, G. Li, dan B. Liu, “A survey of automatic sarcasm detection: Fundamental theories, formulation, datasets, detection methods, and opportunities,” Neurocomputing, vol. 578, hlm. 127428, 2024. [Daring]. Tersedia: https://doi.org/10.1016/j.neucom.2024.127428
[7] A. Bhat dan A. Chauhan, “Multimodal sarcasm detection: A survey,” dalam 2022 IEEE Delhi Section Conference (DELCON), 2022, hlm. 1–7.
[8] A. Baruah, K. Das, F. Barbhuiya, dan K. Dey, “Context-aware sarcasm detection using BERT,” dalam Proceedings of the Second Workshop on Figurative Language Processing, B. B. Klebanov et al., Eds. Online: Association for Computational Linguistics, Jul. 2020, hlm. 83–87. [Daring]. Tersedia: https://aclanthology.org/2020.figlang-1.12/
[9] P. Parameswaran, A. Trotman, V. Liesaputra, dan D. Eyers, “BERT’s the word: Sarcasm target detection using BERT,” dalam Proceedings of the 19th Annual Workshop of the Australasian Language Technology Association, A. Rahimi et al., Eds. Online: ALTA, Des. 2021, hlm. 185–191. [Daring]. Tersedia: https://aclanthology.org/2021.alta-1.21/
[10] G. Huang, Z. Liu, L. Van Der Maaten, dan K. Weinberger, “Densely connected convolutional networks,” dalam Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, hlm. 4700–4708.
[11] S. Pouyanfar et al., “A survey on deep learning: Algorithms, techniques, and applications,” ACM Computing Surveys, vol. 51, no. 5, Art. 92, hlm. 1–36, Sep. 2019. [Daring]. Tersedia: https://doi.org/10.1145/3234150
[12] J. Devlin, M. Chang, K. Lee, dan K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint, arXiv:1810.04805, 2018. [Daring]. Tersedia: https://doi.org/10.48550/arXiv.1810.04805
[13] F. Schroff, D. Kalenichenko, dan J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” dalam Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, hlm. 815–823.
[14] D. Sandberg, "inception_resnet_v1.py," GitHub repository, 2018. [Daring]. Tersedia: https://github.com/davidsandberg/facenet/blob/master/src/models/ince ption_resnet_v1.py
[15] D. Müllner, “Modern hierarchical, agglomerative clustering algorithms,” 2011. [Daring]. Tersedia: https://arxiv.org/abs/1109.2378
[16] S. I. Serengil and A. Ozpinar, "LightFace: A Hybrid Deep Face Recognition Framework," dalam Proceedings of the 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turki, 2020, hlm. 23–27, doi: 10.1109/ASYU50717.2020.9259802.
[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, dan P. Dollár, “Focal Loss untuk Deteksi Objek Padat,” dalam Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italia, 2017, hlm. 2999– 3007. [Daring]. Tersedia: https://doi.org/10.1109/ICCV.2017.324
[18] S. Barak dan T. Mokfi, “Evaluation and selection of clustering methods using a hybrid group MCDM,” Expert Systems with Applications, vol. 138, hlm. 112817, 2019.