

Perbandingan Kinerja LSTM, Bi-LSTM, dan GRU pada Klasifikasi Judul Berita *Clickbait*

Anas Fikri Hanif, Theopilus Bayu Sasongko, Arif Dwi Laksito

anas.f.hanif@students.amikom.ac.id, theopilus.27@amikom.ac.id, arif.laksito@amikom.ac.id

Universitas Amikom Yogyakarta

Informasi Artikel

Diterima : 17 Jul 2023
Direview : 7 Ags 2023
Disetujui : 17 Ags 2023

Kata Kunci

Clickbait, LSTM, Bi-LSTM, GRU, Perbandingan

Abstrak

Maraknya penggunaan konsep *clickbait* menjadi tantangan bagi para pengguna media sosial. Sering kali mereka tertipu dengan judul sebuah artikel yang berbeda dengan isi artikelnya. Oleh karena itu diperlukan sebuah model yang mampu melakukan klasifikasi terhadap judul *clickbait* maupun *non-clickbait*. Meskipun beberapa penelitian sudah dilakukan untuk membuat sebuah model klasifikasi judul *clickbait*, akan tetapi analisa perbandingan sangat diperlukan untuk menentukan model terbaik yang dapat digunakan dalam klasifikasi judul *clickbait*. Oleh karenanya peneliti melakukan perbandingan terhadap tiga model *deep learning* yang berbeda (LSTM, Bi-LSTM, dan GRU) guna menemukan model terbaik yang dapat menyelesaikan kasus ini dengan memanfaatkan data publik dari penelitian sebelumnya. Hasilnya algoritma GRU merupakan algoritma terbaik yang berhasil mencapai akurasi 97,16%. Tidak hanya itu GRU juga memiliki nilai tinggi dalam beberapa metrik evaluasi lainnya, yaitu *precision* 96,63%, *recall* 97,66%, dan *F1-score* 97,14%. Selain menghasilkan metrik evaluasi yang baik, model GRU juga tergolong cepat dalam melakukan *training* dengan waktu 328 detik.

Keywords

Clickbait, LSTM, Bi-LSTM, GRU, Comparison

Abstrak

Clickbait's widespread usage poses a challenge for social media users. Often, they are deceived by article titles that differ from the actual content. Therefore, a model capable of classifying clickbait and non-clickbait titles is needed. Although several studies have been conducted to create a clickbait title classification model, a comparative analysis is essential to determine the best model for this classification task. Hence, researcher compared three different deep learning models (LSTM, Bi-LSTM, and GRU) to find the most effective one, utilizing publicly available data from previous studies. The results revealed that the GRU algorithm is the best-performing one, achieving an accuracy of 97.16%. Moreover, the GRU model exhibited high values in other evaluation metrics, namely, precision of 96.63%, recall of 97.66%, and an F1-score of 97.14%. In addition to producing good evaluation metrics, the GRU model is also considered fast during training, completing the process in just 328 seconds.

A. Pendahuluan

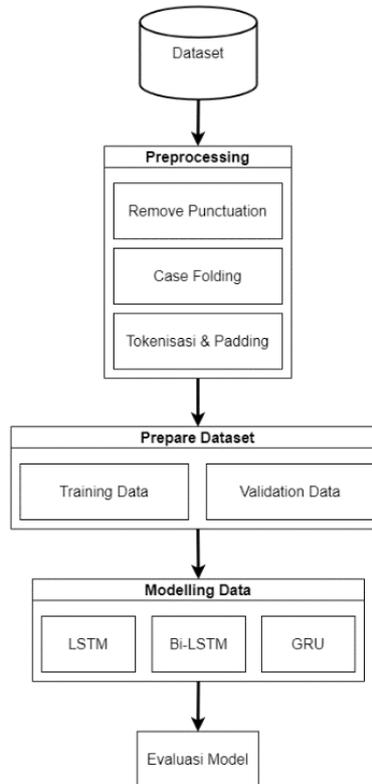
Selama beberapa tahun terakhir perkembangan teknologi yang sangat pesat di seluruh dunia mengubah berbagai aspek kehidupan manusia. Lanskap media dan cara masyarakat mengonsumsi media adalah salah satu aspek yang terdampak oleh perkembangan ini [1]. Sayangnya peralihan masyarakat dari media konvensional ke media digital justru membuat kualitas berita yang ada semakin menurun dengan menjamurnya konsep *clickbait*. *Clickbait* adalah tindakan yang disengaja untuk menjanjikan atau dengan sengaja salah mengartikan judul, citra, maupun kombinasi lain di media sosial yang membuat para pembaca memiliki harapan yang berlebih ketika membaca sebuah cerita di situs web [2]. Meskipun penggunaan *clickbait* di media tidak melanggar undang-undang negara, beberapa kelompok masyarakat marah karena merasa disesatkan oleh *headline* yang tidak sesuai dengan isi berita [3]. Walaupun beberapa model pendeteksian *clickbait* sudah diciptakan, akan tetapi pemilihan model dan algoritma terbaik masih menjadi permasalahan untuk mendapatkan akurasi yang tinggi yang dapat meningkatkan keyakinan pengguna ketika menggunakan sebuah model *machine learning* maupun *deep learning*.

Dalam penelitian sebelumnya, sudah banyak dijumpai penggunaan beberapa model dari *machine learning* maupun *deep learning* dalam klasifikasi judul berita *clickbait*. Pada pemodelan *machine learning* model yang digunakan seperti Logistic Regression, Naïve Bayes dan Random Forest dengan melalui proses TF-IDF berhasil memperoleh akurasi pada kisaran 64% - 66% dalam pengklasifikasian judul berita berbahasa Inggris dan Turki [4]. Penggunaan model Support Vector Machine dan Logistic Regression menghasilkan akurasi pada kisaran 78% - 79% [5]. Penggunaan model *machine learning* lainnya seperti XGBoost (eXtreme Gradient Boosting) Classifier berhasil memperoleh akurasi sebesar 81,2% dalam pengklasifikasian judul *clickbait* dari media sosial Twitter [6]. Sedangkan pada penerapan *deep learning* penggunaan model LSTM berhasil memperoleh akurasi 97% dengan mengambil data dari judul-judul berita di platform Reddit [7]. Penelitian lain memanfaatkan pemodelan LSTM mendapatkan akurasi 93% dan Bi-LSTM mendapatkan akurasi 97% untuk klasifikasi judul berita *clickbait* berbahasa Turki [8]. Model LSTM-RNN juga pernah digunakan untuk memprediksi judul *clickbait* berbahasa Indonesia dan berhasil memperoleh akurasi sebesar 82% [9].

Berdasarkan beberapa penelitian yang sudah dilakukan dalam klasifikasi judul berita *clickbait* terlihat bahwa akurasi yang diperoleh dari pemodelan menggunakan *deep learning* lebih baik daripada pemodelan *machine learning*. Hal ini dikarenakan *deep learning* bekerja dengan meniru cara manusia memperoleh pengetahuan tertentu [10]. Selain itu metode klasifikasi teks tradisional berdasarkan *machine learning* juga memiliki banyak kelemahan seperti ledakan dimensi, persebaran data, kemampuan generalisasi yang terbatas dan lain sebagainya [11]. Oleh karenanya, peneliti mengembangkan beberapa pengoptimalan dalam *deep learning* kemudian menyajikan analisa perbandingan *accuracy*, *precision*, *recall* dan *F1-score* dari tiga arsitektur yang berbeda yaitu LSTM, Bi-LSTM, dan GRU. Peneliti berharap dengan pengaplikasian tiga model *deep learning* yang berbeda dan menggunakan dataset yang sama maka pembaca akan lebih mudah mendapatkan gambaran mengenai model *deep learning* mana yang memiliki performa terbaik dalam pengklasifikasian judul *clickbait*.

B. Metode Penelitian

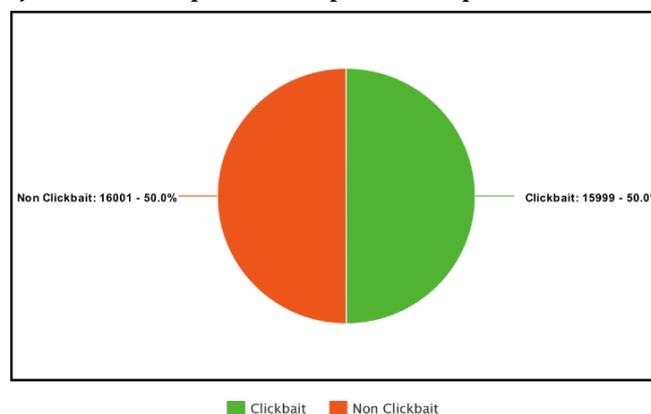
Metodologi penelitian diaplikasikan melalui beberapa tahapan. Tahapan-tahapan tersebut meliputi tahap pengumpulan data, *preprocessing* data, *prepare* data, *modelling* data, dan evaluasi model. Detail dari alur metode penelitian ini ditunjukkan pada Gambar 1 dan dijelaskan pada bagian selanjutnya.



Gambar 1. Alur Metode Penelitian

1. Sumber Data

Penelitian ini memanfaatkan dataset yang sebelumnya pernah digunakan pada penelitian serupa yaitu penelitian yang dilakukan oleh [12]. Dataset ini memuat 32.000 judul artikel berbahasa Inggris yang dikumpulkan dari beberapa situs berita. Detail jumlah data pada setiap label dapat dilihat dari Gambar 2.



Gambar 2. Jumlah Data Tiap Label

Dari Gambar 2 dapat kita lihat bahwa dataset tidak memerlukan proses *balancing* karena dua label dalam dataset memiliki jumlah yang relatif seimbang yaitu 15.999 data *clickbait* dan 16.001 untuk data *non-clickbait*. Untuk judul artikel berlabel *non-clickbait* didapatkan dari situs berita The Hindu, WikiNews, The Guardian, dan New York Times. Sementara untuk judul artikel berlabel *clickbait* didapatkan dari situs berita BuzzFeed, ViralStories, Upworthy, Thatscoop, ViralNova, dan Scoopwhoop, dan. Sampel dataset dapat dilihat pada Tabel 1.

Tabel 1. Sampel Data

Headline	Label
If Disney Princesses Were From Florida	Clickbait
Guys Try Pinterest Leaf Hacks	Clickbait
Does Coffee Make You Poop	Clickbait
Tell Us About Yourself(ie): Erica Ash	Clickbait
When You Realize Every Guy Is Taken	Clickbait
9 Times I Cried	Clickbait
16 Signs You Are Too Stubborn To Live	Clickbait
The Scariest Halloween Candy Taste Test	Clickbait
The 22 Best Adele Memes	Clickbait
Which Type Of Swearer Are You	Clickbait
Haitian earthquake: in pictures	Not clickbait
In Iran, Khatami Loses a Key Backer	Not clickbait
Taiwan's cabinet resigns	Not clickbait
40 injured after attacks in Thailand	Not clickbait
Albanian girl murdered in tangle of crime	Not clickbait
Higher Prices for Popular Comic Books	Not clickbait
U.N. Council May Rebuke North Korea	Not clickbait
Rice angry after abuse of aides in Sudan	Not clickbait
St. Michael and Rice Win Class AA Titles	Not clickbait
Bainimarama sworn in as Fiji caretaker PM	Not clickbait

Nantinya dataset tersebut akan dibagi menjadi *train* data dan *validation* data dengan rasio 90:10 sehingga *train* data berjumlah 28.800 data dan *validation* data berjumlah 3.200 data.

2. Preprocessing Data

Preprocessing dataset terbagi menjadi empat tahapan. Tahapan pertama yaitu pembuangan tanda baca (*punctuation removal*), dilanjutkan dengan penyeragaman semua karakter menjadi huruf kecil (*case folding*), serta tokenisasi dan *padding*.

a) Punctuation Removal

Dalam tahapan ini tanda baca seperti (?!, / = + - \> <; “() {} [].: |) akan ditimpa dengan spasi. Menurut [9], penghapusan tanda baca dengan menggantinya dengan spasi harus dilakukan karena tanda baca diabaikan selama pelatihan. Jadi, dengan hilangnya tanda baca, proses pelatihan menjadi lebih cepat dan mudah.

b) Case Folding

Case folding dilakukan dengan cara mengonversi kata-kata yang memiliki karakter huruf kapital menjadi huruf kecil agar semua data seragam dan tidak mengandung huruf kapital.

c) Tokenisasi dan Padding

Tokenisasi adalah langkah dalam *preprocessing* yang bertujuan untuk mengubah teks agar menjadi *input* yang dapat diterima oleh model. Tokenisasi dilakukan dengan cara memisahkan sebuah kalimat menjadi kata-kata kemudian mengonversi kata-kata tersebut menjadi angka. Setelah tokenisasi dilakukan, langkah selanjutnya adalah melakukan *padding* agar semua data yang dimasukkan ke dalam model memiliki panjang yang sama. Pada penelitian ini, proses tokenisasi dan *padding* dilakukan dengan menggunakan fungsi `Tokenizer()` dan `pad_sequence()` yang disediakan oleh Tensorflow.

3. Algoritma LSTM

Long Short-Term Memory (LSTM) adalah sebuah algoritma yang merupakan pengembangan dari algoritma Recurrent Neural Network (RNN). Pengembangan ini didasari karena masalah *vanishing gradient* dan *exploding gradient* [13] yang terjadi ketika RNN memproses data *long sequence* sehingga RNN kesulitan dalam mempelajari *long-time dependencies* yang memiliki beberapa langkah yang panjang [14]. Masalah-masalah ini berdampak pada menurunnya akurasi pada RNN.

Berdasarkan beberapa masalah di atas, LSTM hadir dengan membawa konsep *memory cell* dalam menyimpan informasi jangka panjang [15]. Persamaan (1) merupakan rumus hasil akhir dari algoritma LSTM di mana h_t adalah *final output*, o_t adalah nilai *output gate*, dan C_t adalah nilai *memory cell state*.

$$h_t = o_t \times \tanh(C_t) \quad (1)$$

4. Algoritma Bi-LSTM

Bidirectional Long Short-Term Memory (BiLSTM) merupakan pengembangan lebih lanjut dari algoritma LSTM. Kelemahan LSTM yang hanya melakukan proses terhadap data secara satu arah memicu lahirnya algoritma BiLSTM. Secara teknis algoritma ini menjalankan dua unit algoritma LSTM dalam satu waktu yaitu satu unit ke arah *forward* dan satu unit ke arah *backward* [16]. Mengingat cara kerja yang sama dengan LSTM maka rumus dari algoritma BiLSTM dapat dilihat pada persamaan (2) di mana h_t adalah *final output*.

$$h_t^{BiLSTM} = h_t^{Forward} \oplus h_t^{Backward} \quad (2)$$

5. Algoritma GRU

Gated Recurrent Unit merupakan bentuk penyederhanaan dari algoritma LSTM. Algoritma ini menggabungkan *input gate* dan *forget gate* dari algoritma LSTM ke dalam sebuah *update gate* [17]. Dengan penggabungan *gate* ini, algoritma GRU dapat berjalan dengan lebih cepat daripada LSTM [18]. Algoritma ini juga dirancang agar bekerja lebih baik daripada LSTM pada data dengan jumlah yang sedikit [19].

Rumus dari algoritma ini dapat dilihat pada persamaan (3) dengan rincian h_t adalah *final output*, z_t adalah *update gate* pada waktu t , h_{t-1} adalah *hidden state* pada waktu sebelumnya, dan \tilde{h}_t adalah *hidden state candidate*.

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \quad (3)$$

6. Evaluasi Model

Pada tahap ini ketiga model yang telah diimplementasikan pada dataset akan dievaluasi performanya. Karena dataset yang digunakan hanya memiliki dua label (biner), maka evaluasi performa model akan menggunakan empat komponen yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Keempat komponen ini akan digunakan untuk menemukan *accuracy*, *precision*, *recall* dan *F1-score* dari masing-masing model dengan perumusan yang tertera pada persamaan (4) hingga persamaan (7).

$$Accuracy = \frac{(TP + TN)}{TP + FP + FN + TN} \quad (4)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (5)$$

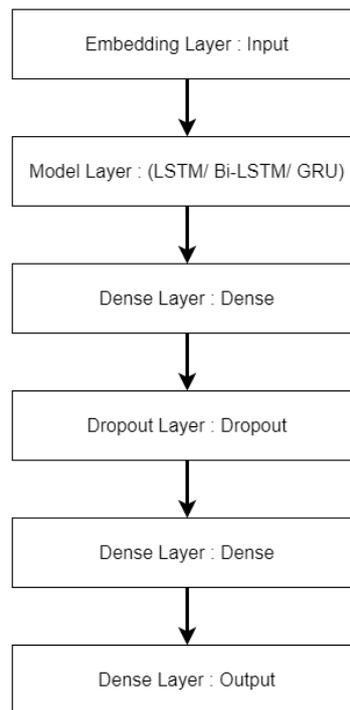
$$Recall = \frac{TP}{(TP+FN)} \quad (6)$$

$$F1 - score = 2 \cdot \frac{(Precision \times Recall)}{(Precision+Recall)} \quad (7)$$

Ketika *accuracy*, *precision*, *recall* dan *F1-score* dari ketiga model telah ditemukan, maka langkah selanjutnya adalah membandingkan model mana yang memiliki performa terbaik. Selain itu waktu yang dibutuhkan dalam melakukan proses *training* data juga akan menjadi pertimbangan mengingat waktu untuk melakukan *training* juga termasuk salah satu poin yang penting. Mengenai informasi perangkat yang digunakan dalam melakukan proses *training* akan dijelaskan pada bagian selanjutnya.

7. Implementasi Model

Penelitian ini memanfaatkan tiga model algoritma *deep learning* yang berbeda yaitu LSTM, Bi-LSTM, dan GRU dalam melakukan klasifikasi judul *clickbait*. Meskipun menggunakan beberapa algoritma yang berbeda, akan tetapi secara garis besar arsitektur yang digunakan sama. Ketiga model tersebut menggunakan satu *embedding layer* sebagai *input layer*, satu *model layer*, dua *dense layer* sebagai *hidden layer*, dan satu *dense layer* sebagai *output layer*. Selain itu, penelitian ini juga memanfaatkan *dropout layer* [10] dan *callback* [9] guna mencegah terjadinya *overfitting* pada model. Penggunaan *layer* yang minim ini dipilih agar proses pengolahan data tidak memakan banyak waktu [18]. Detail dari penggunaan *layer* pada implementasi model terdapat pada Gambar 3.



Gambar 3. Penggunaan *Layer* pada Model

Fungsi aktivasi yang digunakan pada dua *dense layer* adalah ReLu yang memang sering kali digunakan pada *hidden layer* dalam arsitektur *deep learning* akhir-akhir ini. Sementara itu pada satu *dense layer* terakhir, fungsi aktivasi yang digunakan adalah sigmoid karena fungsi ini terbukti memiliki performa yang baik diterapkan pada *output layer* untuk melakukan prediksi probabilitas dalam kasus klasifikasi biner [20]. Dari sisi *loss function*, penelitian ini menggunakan *binary_crossentropy* [9] mengingat dataset yang digunakan hanya memiliki dua label saja. Penelitian ini juga memanfaatkan fungsi optimasi ‘Adam’ yang terbukti sebagai fungsi *optimizer* terbaik dalam beberapa kasus dan dapat mengungguli metode adaptif lainnya seperti *adadelta*, *adagrad*, dan lain-lain [13]. Selanjutnya untuk melakukan evaluasi, penelitian ini memanfaatkan *metric* “accuracy” [9] untuk menghitung akurasi model dari *train data* dan *validation data*.

Proses *training* data dengan menggunakan beberapa model di atas dijalankan pada sebuah *hardware* komputer sebagaimana pada Tabel 2.

Tabel 2. Spesifikasi *Hardware*

Komponen	Spesifikasi
Processor	Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz
RAM (DDR4)	16 GB
SSD	1500 GB
GPU Model	NVIDIA GeForce GTX 1650 12 GB

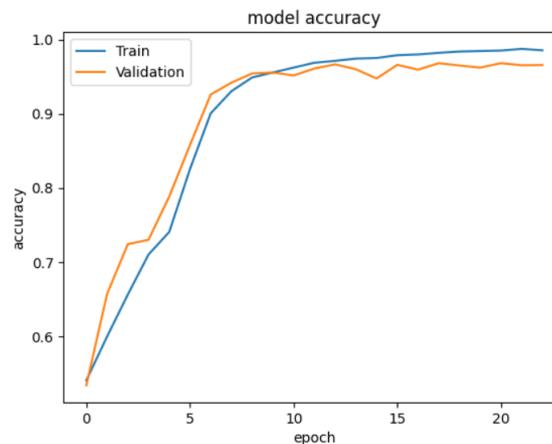
Spesifikasi perangkat keras pada Tabel 2 digunakan sebagai acuan estimasi lama waktu proses *training* dari setiap model. Proses *training* sangat mungkin berlangsung lebih cepat maupun lebih lambat jika dijalankan pada perangkat keras yang memiliki spesifikasi berbeda.

C. Hasil dan Pembahasan

Setelah melakukan beberapa pra pemrosesan pada dataset seperti yang telah dijelaskan, peneliti kemudian mengaplikasikan tiga model yang telah dipilih yaitu model LSTM, Bi-LSTM, dan GRU. Dalam pengaplikasian model, peneliti tidak menentukan jumlah *epochs* secara spesifik melainkan menggunakan bantuan fitur *EarlyStopping* yang dimiliki oleh *library* Keras yang merupakan bagian dari TensorFlow. Fitur ini berfungsi untuk mencegah terjadinya *overfitting* [21] dengan cara menghentikan proses *training* apabila beberapa parameter terpenuhi. Dalam menggunakan fitur ini, ditentukan tiga parameter utama yaitu parameter *monitor*, *patience*, dan *min_delta*. Untuk parameter *monitor* sendiri digunakan *value* berupa *val_accuracy* yang berfungsi untuk memantau metrik akurasi dari data validasi. Kemudian pada parameter *patience* menggunakan *value* 10 yang berarti *epoch* akan tetap berlanjut selama 10 kali tanpa adanya peningkatan pada metrik *val_accuracy*. Selanjutnya pada parameter terakhir yaitu *min_delta* menggunakan *value* 0,01 yang berarti sebuah *epoch* baru dikatakan mengalami peningkatan jika metrik *val_accuracy* meningkat minimal 0,01 dari *epoch* sebelumnya.

1. Hasil Model LSTM

Model pertama yang digunakan untuk melakukan proses *training* adalah model LSTM (Long Short-Term Memory). Dengan memanfaatkan beberapa *layer* dari model Sequential, penulis berhasil memperoleh akurasi yang cukup tinggi yaitu 96,5%. Akurasi ini didapatkan ketika proses *training* mencapai *epoch* ke-23 di mana pada *epoch* tersebut proses *training* harus dihentikan karena metrik *val_accuracy* telah memenuhi beberapa persyaratan yang sebelumnya ditentukan dalam *callback* *EarlyStopping*.



Gambar 4. Akurasi Model LSTM pada Setiap Epoch

Gambar 4 memperlihatkan akurasi yang didapatkan model saat proses *training* mulai dari *epoch* pertama hingga terakhir. Terlihat dari gambar tersebut akurasi model dari data *training* dan *validation* tidak memiliki perbedaan yang cukup signifikan. Pada *epoch* ke-23 yang merupakan *epoch* terakhir, akurasi dari data *training* mencapai 98,5% sementara akurasi dari data *validation* mencapai 96,2%. Hal ini menandakan bahwa model LSTM tidak mengalami *overfitting* karena model masih bisa beradaptasi dengan baik ketika dihadapkan pada data *validation*.

Tabel 3. *Confusion Matrix* Model LSTM

	True Label Positive	True Label Negative
Predicted Label Positive	1564	36
Predicted Label Negative	74	1526

Dari *confusion matrix* pada Tabel 3, maka peneliti dapat mengetahui komponen *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Selanjutnya komponen-komponen ini akan digunakan untuk melakukan evaluasi model LSTM yang meliputi *accuracy*, *precision*, *recall*, dan *F1-score* yang dapat dilihat pada Tabel 4.

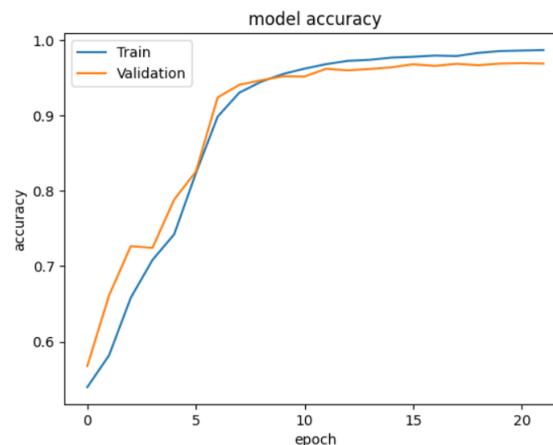
Tabel 4. Evaluasi Model LSTM

Accuracy	Precision	Recall	F1-Score
96,56%	97,75%	95,48%	96,60%

Berdasarkan data evaluasi model LSTM pada Tabel 4, maka bisa dikatakan model ini termasuk model yang cukup baik di mana *accuracy*, *precision*, *recall*, dan *F1-score* memiliki nilai yang cukup tinggi. Dalam pembuatan model ini, proses *training* membutuhkan waktu selama 5 menit 2 detik yang tergolong cukup cepat mengingat jumlah data yang tidak sedikit.

2. Hasil Model Bi-LSTM

Model selanjutnya yang digunakan dalam proses *training* adalah model dengan algoritma Bi-LSTM (Bidirectional Long Short-Term Memory). Pada dasarnya model ini bekerja mirip seperti model LSTM yang sebelumnya telah diimplementasikan pada data. Algoritma Bi-LSTM dengan menggunakan dua unit yaitu *forward* dan *backward* dalam proses *training* terbukti memperlihatkan hasil akurasi yang sedikit lebih tinggi dibandingkan model LSTM biasa. Akurasi yang dihasilkan oleh model Bi-LSTM adalah 96,8%. Akurasi ini didapatkan pada *epoch* ke-22 di mana pada *epoch* tersebut proses *training* dihentikan oleh fungsi *callback*. Gambaran mengenai akurasi yang diperoleh pada tiap *epoch* dapat dilihat pada Gambar 5.

**Gambar 5.** Akurasi Model Bi-LSTM pada Setiap *Epoch*

Dari diagram pada Gambar 5 dapat diketahui bahwa meskipun akurasi dari data *validation* sempat memiliki perbedaan yang cukup signifikan dengan data *training* pada *epoch* awal, akan tetapi mendekati *epoch* ke-5 akurasi dari kedua data tersebut relatif tidak berbeda jauh. Memasuki *epoch* ke-5 tersebut kedua akurasi baik akurasi data *train* maupun *validation* berjalan beriringan. Memasuki *epoch* ke-10 kedua akurasi tidak mengalami banyak peningkatan dan cenderung stagnan hingga akhirnya pada *epoch* ke-22 proses *training* dihentikan oleh fungsi *callback*. Pada *epoch* ke-22 ini akurasi data *train* mencapai 98,6% sementara akurasi dari data *validation* mencapai 96,8%.

Tabel 5. *Confusion Matrix* Model Bi-LSTM

	True Label Positive	True Label Negative
Predicted Label Positive	1559	41
Predicted Label Negative	59	1541

Dari *confusion matrix* model Bi-LSTM pada Tabel 5 kemudian dapat dikalkulasikan *accuracy*, *precision*, *recall*, dan *F1-score* untuk melakukan evaluasi model. Evaluasi ini dapat dilihat pada Tabel 6.

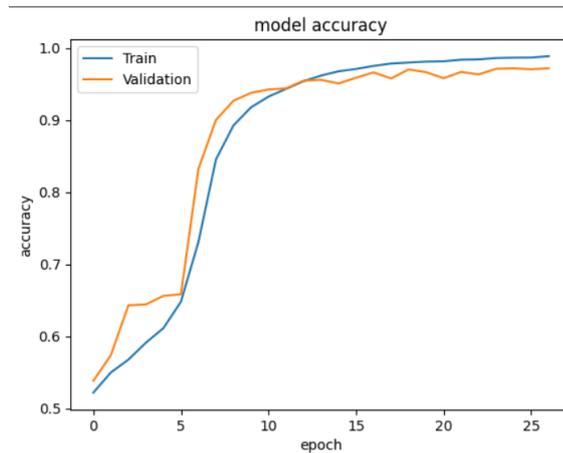
Tabel 6. Evaluasi Model Bi-LSTM

Accuracy	Precision	Recall	F1-Score
96,88%	97,44%	96,35%	96,89%

Terbukti dengan menggunakan mekanisme gandanya membuat *accuracy*, *precision*, *recall*, dan *F1-score* dari algoritma Bi-LSTM sedikit lebih tinggi daripada metrik evaluasi yang didapatkan pada algoritma LSTM. Meski demikian perbedaan yang dihasilkan oleh kedua model ini tidak terlalu jauh berbeda dan keduanya masih bisa dikatakan sebagai model yang baik karena tidak mengalami *overfitting* dan dapat beradaptasi dengan data baru. Proses training pada model ini sendiri memakan waktu 7 menit 23 detik.

3. Hasil Model GRU

Model terakhir yang digunakan untuk melatih data adalah model GRU (Gated Recurrent Unit). Model ini memiliki prinsip kerja yang sedikit berbeda dari dua model yang sebelumnya telah diaplikasikan untuk melatih data. Tujuan awal pembuatan model ini adalah agar mampu bekerja lebih baik daripada arsitektur model LSTM pada data dengan jumlah yang sedikit. Pada pelatihan kali ini dengan menggunakan data berjumlah 28.800 menunjukkan bahwa performa model sedikit lebih tinggi daripada dua model yang sebelumnya telah diimplementasikan. Model GRU memperoleh akurasi sebesar 97,1% di mana akurasi ini berada sedikit di atas model Bi-LSTM, bahkan LSTM. Akurasi ini diperoleh saat pelatihan model mencapai *epoch* ke-27 ketika pelatihan dihentikan oleh fungsi *callback*. Pada Gambar 6, dapat dilihat akurasi yang dihasilkan oleh model GRU dalam setiap *epoch*.



Gambar 6. Akurasi Model GRU pada Setiap *Epoch*

Pada Gambar 6 dapat dilihat bahwa akurasi yang didapat dari proses pelatihan data menggunakan algoritma GRU berjalan cukup baik. Akurasi dari data *train* dan *validation* relatif tidak berbeda jauh setelah melalui *epoch* ke-5. Saat proses *training* berhenti pada *epoch* ke-27, akurasi dari kedua data ini juga memiliki perbedaan yang tidak begitu jauh dengan akurasi pada data *train* sebesar 98,8% sedangkan akurasi pada data *validation* sebesar 97,1%. Perbedaan yang tidak begitu besar ini membuktikan suksesnya fungsi *callback* untuk mencegah terjadinya *overfitting* pada model.

Tabel 7. *Confusion Matrix* Model GRU

	True Label Positive	True Label Negative
Predicted Label Positive	1546	54
Predicted Label Negative	37	1563

Pelatihan dengan menggunakan model GRU menghasilkan *confusion matrix* seperti pada Tabel 7. Dari *confusion matrix* ini kemudian dilakukan perhitungan untuk menemukan komponen metrik evaluasi berupa *accuracy*, *precision*, *recall*, dan *F1-score*. Selanjutnya hasil perhitungan dari keempat komponen ini dapat dilihat pada Tabel 8.

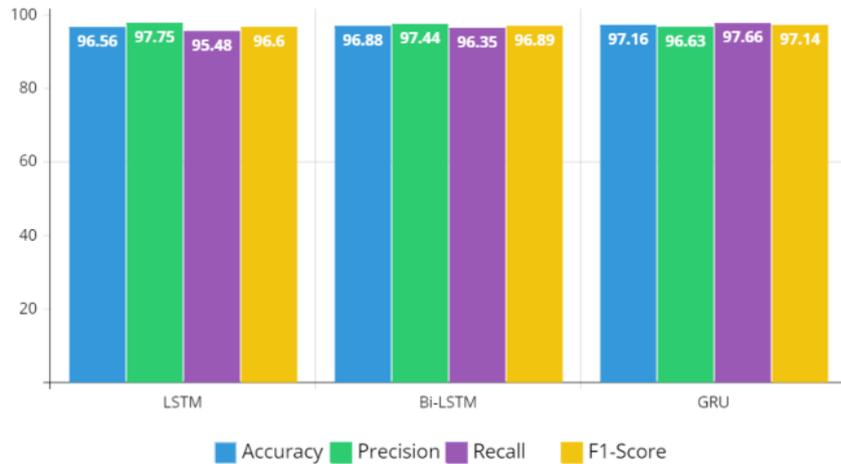
Tabel 8. Evaluasi Model GRU

Accuracy	Precision	Recall	F1-Score
97,16%	96,63%	97,66%	97,14%

Tabel 8 memperlihatkan bagaimana algoritma GRU mampu mencapai akurasi di atas 97% ketika dua algoritma sebelumnya hanya mencapai akurasi di bawah 97%. Selain itu algoritma GRU juga memiliki nilai *precision*, *recall*, dan *F1-score* yang cukup tinggi mengindikasikan bahwa model berjalan dengan baik. Model GRU juga tergolong cepat dalam hal waktu *training* di mana model ini hanya membutuhkan waktu 5 menit 28 detik untuk menyelesaikan keseluruhan proses *training*.

4. Perbandingan Kinerja Model LSTM, Bi-LSTM, dan GRU

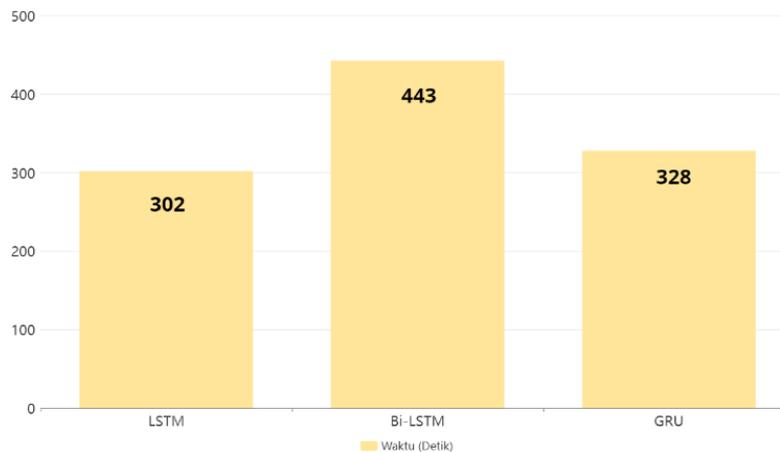
Seperti apa yang telah dijelaskan sebelumnya evaluasi ketiga model didasarkan pada empat metrik evaluasi yaitu *accuracy*, *precision*, *recall* dan *F1-score*. Selain itu ketiga model juga akan dievaluasi berdasarkan lama waktu yang dibutuhkan untuk melakukan proses *training* pada data. Perbandingan metrik evaluasi dari ketiga model dapat dilihat pada gambar 7.



Gambar 7. Perbandingan Metrik Evaluasi dari Ketiga Model

Jika melihat dari Gambar 7 dapat diketahui bahwa ketiga model memiliki performa yang cukup berimbang dengan selisih kurang dari 3% pada setiap metrik. Akan tetapi metrik evaluasi yang dihasilkan oleh model GRU adalah metrik dengan keseluruhan nilai terbaik meskipun algoritma GRU terbilang cukup lemah dalam metrik *precision* dengan hanya mencapai angka 96,63%. Angka ini sedikit di bawah metrik *precision* yang dimiliki oleh model LSTM dan Bi-LSTM yang mencapai 97,75% dan 97,44%. Akan tetapi jika melihat tiga metrik lainnya yaitu *accuracy*, *recall*, dan *F1-score* algoritma GRU mendominasi dengan berhasil memperoleh angka tertinggi pada ketiga metrik tersebut.

Namun, perbandingan kinerja tiga model tersebut juga dilakukan berdasarkan waktu yang diperlukan untuk melakukan *training* terhadap data. Detail mengenai waktu yang dibutuhkan setiap model untuk melakukan *training* dapat dilihat pada Gambar 8.



Gambar 8. Waktu *Training* Setiap Model

Dari Gambar 8 dapat dilihat bahwa waktu paling cepat diperoleh oleh model LSTM dengan hanya memerlukan waktu 302 detik untuk menyelesaikan seluruh proses *training*. Sementara waktu terlama dimiliki oleh algoritma Bi-LSTM dengan waktu 443 detik. Perbedaan waktu yang cukup jauh antara LSTM dan Bi-LSTM sangat wajar terjadi mengingat algoritma Bi-LSTM secara prinsip melakukan dua kali proses yang dilakukan oleh algoritma LSTM biasa. Sementara algoritma GRU yang diharapkan menjadi penyederhana algoritma LSTM justru memiliki waktu sedikit lebih lama daripada LSTM yaitu berada di angka 328 detik.

D. Simpulan

Clickbait adalah tindakan yang disengaja untuk menjanjikan atau dengan sengaja salah mengartikan judul, citra, maupun kombinasi lain di media sosial yang membuat para pembaca memiliki harapan yang berlebih ketika membaca sebuah cerita di situs web. Berdasarkan dataset yang berisi judul *clickbait* dan *non-clickbait* yang berjumlah 32.000 data dengan perbandingan kelas data 50:50, peneliti berhasil mengaplikasikan 3 algoritma *deep learning* untuk melakukan klasifikasi terhadap judul-judul tersebut apakah masuk ke dalam kategori *clickbait* atau bukan. Ketiga algoritma *deep learning* yang digunakan adalah LSTM, Bi-LSTM, dan GRU. Setelah melakukan proses *training* dan mengevaluasi beberapa parameter dari masing-masing model, maka diperoleh model yang menggunakan algoritma GRU merupakan model terbaik. Model ini memiliki metrik *accuracy* 97,16%, *precision* 96,63%, *recall* 97,66%, dan *F1-score* 97,14%. Keempat metrik yang dihasilkan oleh algoritma GRU ini merupakan metrik terbaik secara keseluruhan jika dibandingkan dengan dua model lainnya. Sementara itu dari sisi waktu yang dibutuhkan untuk melakukan *training* data, model dengan algoritma GRU memerlukan waktu 328 detik. Meskipun bukan yang tercepat, akan tetapi waktu yang diperlukan oleh model GRU ini hanya memiliki selisih 26 detik dari LSTM yang merupakan model dengan waktu *training* tercepat. Meskipun telah diketahui bahwa algoritma GRU merupakan yang terbaik dalam melakukan klasifikasi judul *clickbait* menggunakan dataset yang telah disebutkan, akan tetapi peneliti menyarankan untuk melakukan pengembangan dengan melakukan modifikasi pada *layer sequential*, *hyperparameter tuning* pada setiap model, serta menambahkan proses *stemming* maupun *lemmatization* pada data sebelum melakukan pemodelan guna memperoleh hasil yang lebih baik.

E. Ucapan Terima Kasih

Penulis mengucapkan terimakasih kepada Universitas Amikom Yogyakarta dan semua pihak yang telah memberikan dukungan dalam penelitian ini.

F. Referensi

- [1] Y. D. Hadiyat, "Clickbait on Indonesia Online Media," *Journal Pekommas*, vol. 4, no. 1, p. 1, Apr. 2019, doi: 10.30818/jpkm.2019.2040101.
- [2] V. Kumar, D. Khattar, S. Gairola, Y. Kumar Lal, and V. Varma, "Identifying Clickbait: A Multi-Strategy Approach Using Neural Networks *," 2018. doi: 10.1145/3209978.3210144.

-
- [3] O. Solihin, W. Rahmawati, F. Haryati, Y. Mogot, N. Z. Fachrul, and E. A. Waluyo, "TINJAUAN TENTANG CLICKBAIT MEDIA 1," *Jurnal Komunikasi dan Media*, vol. 7, no. 1, 2022, doi: 10.33884/commed.v7i1.6599.
- [4] A. Geckil, A. A. Mungen, E. Gundogan, and M. Kaya, "A clickbait detection method on news sites," in *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018*, Institute of Electrical and Electronics Engineers Inc., Oct. 2018, pp. 932–937. doi: 10.1109/ASONAM.2018.8508452.
- [5] D. M. Daoud and M. S. A. El-Seoud, "An effective approach for clickbait detection based on supervised machine learning technique," *International journal of online and biomedical engineering*, vol. 15, no. 3, pp. 21–32, 2019, doi: 10.3991/ijoe.v15i03.9843.
- [6] A. Elyashar, J. Bendahan, and R. Puzis, "Detecting Clickbait in Online Social Media: You Won't Believe How We Did It," pp. 337–387, Oct. 2017, doi: 10.1007/978-3-031-07689-3_28.
- [7] B. Naeem, A. Khan, O. Mirza, Beg, and H. Mujtaba, "A Deep Learning Framework for Clickbait Detection on Social Area Network using Natural Language Cues," 2020. doi: 10.1007/s42001-020-00063-y.
- [8] Ş. Genç and E. Surer, "ClickbaitTR: Dataset for clickbait detection from Turkish news sites and social media with a comparative analysis via machine learning algorithms," *J Inf Sci*, Apr. 2021, doi: 10.1177/01655515211007746.
- [9] B. Siregar, I. Habibie, E. B. Nababan, and Fahmi, "Identification of Indonesian clickbait news headlines with long short-term memory recurrent neural network algorithm," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, May 2021. doi: 10.1088/1742-6596/1882/1/012129.
- [10] R. N. S. Amriza and D. Supriyadi, "Komparasi Metode Machine Learning dan Deep Learning untuk Deteksi Emosi pada Text di Sosial Media," *JUPITER (JURNAL PENDIDIKAN TEKNIK ELEKTRO)*, vol. 13, no. 2, pp. 130–139, 2021, doi: 10.5281/3603.jupiter.2021.10.
- [11] H. Wu, Y. Liu, and J. Wang, "Review of text classification methods on deep learning," *Computers, Materials and Continua*, vol. 63, no. 3. Tech Science Press, pp. 1309–1321, Apr. 01, 2020. doi: 10.32604/CMC.2020.010172.
- [12] A. Chakraborty, B. Paranjape, S. Kakarla, and N. Ganguly, "Stop Clickbait: Detecting and Preventing Clickbaits in Online News Media," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2016, pp. 9–16. doi: 10.1109/ASONAM.2016.7752207.
- [13] N. K. Manaswi, *Deep Learning with Applications Using Python*. Apress, 2018. doi: 10.1007/978-1-4842-3516-4.
- [14] H. Chung and K. S. Shin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," *Sustainability (Switzerland)*, vol. 10, no. 10, Oct. 2018, doi: 10.3390/su10103765.
- [15] A. Hanifa and S. Akbar, "Detection of Unstable Approaches in Flight Track with Recurrent Neural Network," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, 2018, pp. 735–740. doi: 10.1109/icoiact.2018.8350754.

-
- [16] H. Elfaik and E. H. Nfaoui, "Deep Bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic Text," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 395–412, Jan. 2021, doi: 10.1515/jisys-2020-0021.
- [17] A. Dutta, S. Kumar, and M. Basu, "A Gated Recurrent Unit Approach to Bitcoin Price Prediction," *Journal of Risk and Financial Management*, vol. 13, no. 2, p. 23, Feb. 2020, doi: 10.3390/jrfm13020023.
- [18] W. Hastomo, A. S. B. Karno, N. Kalbuana, E. Nisfiani, and L. ETP, "Optimasi Deep Learning untuk Prediksi Saham di Masa Pandemi Covid-19," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 7, no. 2, pp. 133–140, 2021, doi: 10.26418/jp.v7i2.47411.
- [19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 2014, [Online]. Available: <http://arxiv.org/abs/1412.3555>
- [20] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," Nov. 2018, [Online]. Available: <http://arxiv.org/abs/1811.03378>
- [21] P. Schneider, D. Müller, and F. Kramer, "CLASSIFICATION OF VIRAL PNEUMONIA X-RAY IMAGES WITH THE AUCMEDI FRAMEWORK," Oct. 2021, Accessed: May 24, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2110.01017>