

Penerapan *Convolutional Long Short-Term Memory* untuk Klasifikasi Teks Berita Bahasa Indonesia

(*Convolutional Long Short-Term Memory Implementation for Indonesian News Classification*)

Yudi Widhiyana¹, Transmissia Semiawan², Ilham Gibran Achmad Mudzakir³, Muhammad Randi Noor⁴

Abstract—Text classification is now a well-studied field, particularly in Natural Language Processing (NLP). The text classification can be carried out using various methods, one of which is deep learning. Deep learning methods such as RNN, CNN, and LSTM are the most frequent methods used for text classification. This research aims to analyze the implementation of two deep learning methods combination, namely CNN and LSTM (C-LSTM), to classify Indonesian news texts. News texts used as data in this study were collected from Indonesian news portals. The obtained data were then divided into three categories based on their scope: "National," "International," and "Regional." Three research variables were tested in this study: the number of documents, the batch size value, and the learning rate value of the built C-LSTM. The experimental results showed that the F1-score obtained from the classification results using the C-LSTM method was 93.27%. The F1-score value generated by the C-LSTM method was higher than that of CNN (89.85%) and LSTM (90.87%). In summary, the combination method of two deep learning methods, namely CNN and LSTM (C-LSTM), outperforms CNN and LSTM.

Intisari—Klasifikasi teks saat ini telah menjadi sebuah bidang yang banyak diteliti, khususnya terkait *Natural Language Processing* (NLP). Terdapat banyak metode yang dapat dimanfaatkan untuk melakukan klasifikasi teks, salah satunya adalah metode *deep learning*. RNN, CNN, dan LSTM merupakan beberapa metode *deep learning* yang umum digunakan untuk mengklasifikasikan teks. Makalah ini bertujuan menganalisis penerapan kombinasi dua buah metode *deep learning*, yaitu CNN dan LSTM (C-LSTM). Kombinasi kedua metode tersebut dimanfaatkan untuk melakukan klasifikasi teks berita bahasa Indonesia. Data yang digunakan adalah teks berita bahasa Indonesia yang dikumpulkan dari portal-portal berita berbahasa Indonesia. Data yang dikumpulkan dikelompokkan menjadi tiga kategori berita berdasarkan lingkungannya, yaitu "Nasional", "Internasional", dan "Regional". Dalam makalah ini dilakukan eksperimen pada tiga buah variabel penelitian, yaitu jumlah dokumen, ukuran *batch*, dan nilai *learning rate* dari C-LSTM yang dibangun. Hasil eksperimen menunjukkan bahwa nilai *F1-score* yang diperoleh dari hasil klasifikasi menggunakan metode C-LSTM adalah sebesar 93,27%. Nilai *F1-score* yang dihasilkan oleh metode C-LSTM lebih besar dibandingkan dengan CNN, dengan nilai 89,85%, dan LSTM, dengan nilai 90,87%. Dengan demikian, dapat disimpulkan bahwa kombinasi dua metode *deep*

learning, yaitu CNN dan LSTM (C-LSTM), memiliki kinerja yang lebih baik dibandingkan dengan CNN dan LSTM.

Kata Kunci—*Deep Learning*, CNN, LSTM, Klasifikasi Teks, Teks Berita.

I. PENDAHULUAN

Klasifikasi teks (*text classification*) telah menjadi bidang yang banyak diteliti di bidang *Natural Language Processing* (NLP) [1]. Penerapan klasifikasi teks telah banyak dilihat dalam kehidupan sehari-hari. Salah satu penerapan klasifikasi teks yang dapat ditemui adalah proses pengklasifikasian teks berita berdasarkan kategori tertentu. Saat ini, terdapat dua metode yang paling umum digunakan dalam melakukan klasifikasi teks, yaitu menggunakan metode tradisional (*parametric algorithm*) dan *deep learning* [2]. Dengan berkembangnya teknologi di bidang NLP, proses klasifikasi teks saat ini banyak dikembangkan dengan memanfaatkan teknologi *neural network*, salah satunya adalah *deep learning*. *Deep learning* terbukti dapat melakukan proses klasifikasi teks dengan sangat baik [3]–[5].

Berbagai algoritme *deep learning* telah dicoba untuk melakukan klasifikasi berita berdasarkan kategori tertentu. Metode *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM) adalah dua algoritme yang umum digunakan untuk mengenali data yang memiliki keterkaitan di dalamnya [6]. Kendati demikian, metode *Convolutional Neural Network* (CNN), yang umumnya digunakan untuk pemrosesan citra, juga memperlihatkan kinerja yang baik dalam mengklasifikasi teks [3].

Referensi [4], [7] mengombinasikan dua buah arsitektur *deep learning*, yaitu CNN dan LSTM untuk melakukan klasifikasi teks berita. Kombinasi dua arsitektur ini diberi nama *Convolutional Long Short-Term Memory* (C-LSTM). Referensi [4], [7] memperlihatkan akurasi C-LSTM yang lebih baik daripada CNN dan LSTM saat dibuat terpisah. *Dataset* yang digunakan adalah berita dengan bahasa Inggris dan bahasa Mandarin. Kedua penelitian ini memunculkan sebuah pertanyaan mengenai pemanfaatan C-LSTM untuk melakukan klasifikasi berita berupa teks dalam bahasa Indonesia. Oleh karena itu, dalam makalah ini dilakukan eksperimen terkait penerapan C-LSTM untuk melakukan klasifikasi berita teks berbahasa Indonesia.

II. KLASIFIKASI TEKS

Klasifikasi teks adalah proses pengelompokan teks atau dokumen ke dalam beberapa kategori atau kelas [1]. Secara

^{1,2,3,4} Jurusan Teknik Komputer dan Informatika Politeknik Negeri Bandung, Jln. Gegerkalong Hilir, Ds. Ciwaruga, Kec. Parompong, Kab. Bandung Barat 40559 INDONESIA (telp: 022-2015721; fax: 022-2013889; e-mail: ¹goezthel@jtk.polban.ac.id; ²transmissia@jtk.polban.ac.id; ³ilham.gibran.tif417@polban.ac.id; ⁴muhammad.randi.tif417@polban.ac.id)

umum, proses klasifikasi teks melibatkan beberapa tahapan, seperti ekstraksi fitur, *dimension reduction*, *classifier selection*, dan *evaluation* [1]. Telah banyak penelitian yang mencoba membahas proses klasifikasi teks dengan berbagai studi kasus yang berbeda. Studi kasus teks berita adalah salah satu studi kasus yang banyak diteliti saat ini. Metode *deep learning* adalah salah satu metode yang saat ini banyak digunakan untuk melakukan klasifikasi teks [2].

Terdapat sebuah penelitian yang fokus membahas berbagai metode yang dapat digunakan pada klasifikasi teks [1]. Penelitian ini juga membahas secara sederhana tahapan-tahapan yang terdapat pada klasifikasi teks. Terdapat juga penelitian yang melakukan klasifikasi teks berita bahasa Indonesia dengan memanfaatkan metode CNN [3]. Penelitian tersebut menunjukkan bahwa CNN, yang umumnya digunakan untuk menyelesaikan masalah terkait pengolahan citra, dapat juga diterapkan untuk menyelesaikan masalah terkait dengan NLP. CNN yang digunakan memanfaatkan *convolution layer ID* untuk mengekstrak fitur dari dalam berita [3]. CNN yang dibangun dapat melakukan klasifikasi dengan akurasi $\geq 90\%$ pada *test set* yang digunakan.

Telah dilakukan juga penelitian yang sama-sama memanfaatkan CNN untuk melakukan klasifikasi berita bahasa Mandarin [8]. Namun, penelitian tersebut mencoba mengombinasikan CNN dengan *Gated Recurrent Unit* (GRU) dan *Support Vector Machine* (SVM) untuk membentuk sebuah *voting system*. *Voting system* yang dibangun mengklasifikasikan berita berdasarkan hasil *voting* dari ketiga metode yang digunakan. Metode yang dipilih dapat melakukan klasifikasi dengan akurasi sebesar 81% yang dihitung dengan memanfaatkan *macro F1-score*, yaitu nilai *F1-score* yang dihitung dengan *macro-average*.

Referensi [2] mencoba memanfaatkan LSTM untuk mengklasifikasikan berita dengan bahasa Mandarin dan Inggris. LSTM dipilih karena kemampuannya dalam mengenali keterkaitan antar konteks di dalam dokumen teks [2], [4]. Metode yang dipilih mengombinasikan LSTM dengan *hierarchical attentions* untuk meningkatkan akurasi yang dihasilkan oleh LSTM. Penelitian ini juga menambahkan modul *bidirectional* pada LSTM untuk membuat LSTM mengenali keterkaitan konteks setelah dan sebelum kata yang sedang diolah. Metode yang dipilih menghasilkan akurasi $\geq 90\%$ pada tiga *dataset* berita yang digunakan.

Penelitian lain mengombinasikan CNN dan LSTM untuk melakukan klasifikasi teks berita [4], [7]. CNN memiliki kelebihan dalam melakukan ekstraksi fitur karena didukung dengan *convolution layer* yang mampu melakukan ekstraksi fitur dengan lebih baik. Sementara itu, LSTM memiliki kelebihan dalam memperhatikan keterkaitan konteks di dalam dokumen berita.

Referensi [7] mencoba menyusun arsitektur CNN dan LSTM secara *sequential*. Data dimasukkan terlebih dahulu ke dalam LSTM, kemudian diproses lebih lanjut memanfaatkan CNN. Hal ini berbeda dengan penelitian yang mengombinasikan CNN dan LSTM secara paralel [4], sehingga kedua metode sama-sama dimanfaatkan untuk melakukan klasifikasi. Hasil yang diperoleh menunjukkan bahwa kombinasi CNN dan

LSTM menghasilkan akurasi klasifikasi yang lebih baik daripada CNN maupun LSTM yang tidak dikombinasikan [4], [7].

III. PENGEMBANGAN C-LSTM

Bagian ini secara khusus menjelaskan proses pengembangan metode C-LSTM yang digunakan untuk melakukan pengklasifikasian teks. Metode C-LSTM sebenarnya telah diajukan pada penelitian sebelumnya [4]. Namun, makalah ini mengimplementasikan metode C-LSTM untuk teks berita yang berbeda dengan penelitian pada [4]. Pengembangan metode dilakukan dengan arsitektur yang terdapat pada [4], tetapi dengan pengaturan *hyperparameter* yang berbeda.

A. Dataset

Dataset yang digunakan dalam makalah ini adalah berita dalam bentuk teks. Teks berita dikumpulkan dari portal berita Detik, Suara, CNN Indonesia, Tribun, Tempo, dan Kompas. Terdapat 5.000 dokumen berita yang berhasil dikumpulkan. Teks berita yang dikumpulkan kemudian disimpan sebagai *plain text* di dalam *file* dengan format *comma separated value* (CSV). Penyimpanan di dalam *file* CSV dilakukan untuk mempermudah proses pembacaan dokumen berita pada tahap *data labeling* dan *text pre-processing*. Pada tahap *data labeling*, teks berita dikelompokkan ke dalam tiga buah kategori, yaitu “Internasional”, “Nasional”, dan “Regional”.

B. Pre-processing Dokumen Berita

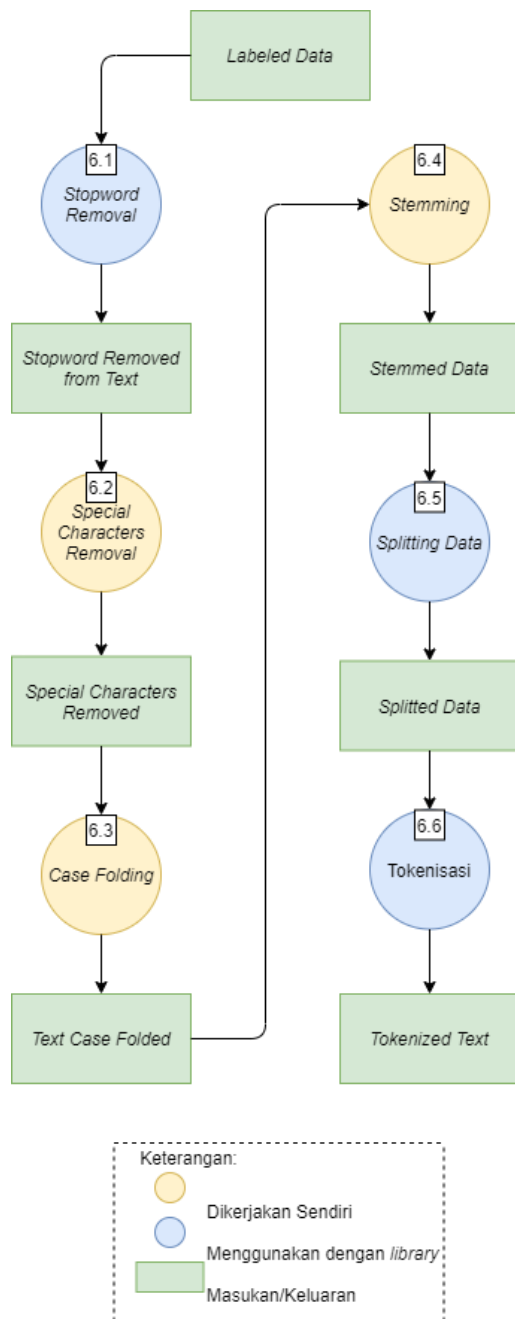
Dokumen berita yang telah dikumpulkan tidak dapat langsung digunakan untuk proses pelatihan *machine learning* yang dibangun. Dokumen berita perlu melalui tahapan *text pre-processing* untuk mempersiapkan dokumen berita sehingga dapat digunakan untuk pelatihan *machine learning*. Tahapan *text pre-processing* bertujuan untuk membersihkan teks berita yang diperoleh.

Pembersihan teks disini memiliki arti menyetarakan format penulisan dokumen berita, sehingga lebih mudah diakses pada saat pelatihan berlangsung. Tahap *text pre-processing* juga dilakukan sebagai usaha meningkatkan kinerja pengklasifikasian C-LSTM, yaitu dengan menyesuaikan dokumen berita yang diperoleh agar mudah dipahami oleh C-LSTM yang dibangun.

Terdapat beberapa tahapan *text processing* yang dilaksanakan, seperti ditunjukkan pada Gbr. 1. Tahapan tersebut yaitu *stopword removal*, *special character removal*, *case folding*, *stemming*, dan *data splitting*. Seluruh tahapan *pre-processing* ini dilakukan setelah dokumen berita diberi label untuk mengidentifikasi masing-masing kategori pada berita.

Stopword removal adalah sebuah tahapan pada *text pre-processing* yang berfungsi menghilangkan kata-kata yang tidak memiliki makna khusus, misalnya kata hubung, seperti ‘dan’, ‘karena’, dan ‘tetapi’. Kata hubung perlu dihilangkan karena sering muncul dan berpotensi memengaruhi kinerja pengklasifikasian C-LSTM.

Special character removal adalah tahapan untuk menghilangkan tanda baca dan karakter-karakter lainnya yang tidak memiliki makna khusus. Sama halnya dengan *stopword* yang sering muncul, karakter-karakter seperti tanda baca juga

Gbr. 1 Tahapan *pre-processing* yang dilakukan.

dapat memengaruhi kinerja pengklasifikasian oleh *machine learning*.

Tahapan *data pre-processing* selanjutnya adalah *case folding*, yang bertujuan untuk menyamaratakan bentuk huruf-huruf pada data teks. Pada makalah ini, karakter yang terdapat di dalam dokumen berita dikonversi ke dalam bentuk *lowercase*.

Stemming adalah proses transformasi setiap kata pada teks berita menjadi bentuk kata dasarnya. Konversi ke dalam bentuk kata dasar bertujuan untuk menyelaraskan pemaknaan dari suatu kata. Selain itu, *stemming* juga dapat mempermudah *machine learning* dalam mengolah dan mengenali kata yang terdapat pada dokumen berita.

Proses terakhir adalah proses *data splitting*, yang berfungsi membagi *dataset* menjadi data latih dan data uji. Pembagian *dataset* ini bertujuan untuk memastikan bahwa berita yang digunakan untuk menguji C-LSTM yang telah dilatih berbeda dengan dokumen yang dimanfaatkan untuk pelatihan. Dengan demikian, perhitungan terhadap kinerja pengklasifikasian dapat dinilai dengan lebih baik. Melalui pemisahan jenis *dataset* ini, dapat juga diperoleh pengetahuan tentang C-LSTM yang dibangun, yaitu mengalami *overfitting* atau tidak. *Overfitting* adalah kondisi saat *machine learning* memiliki kinerja klasifikasi yang baik hanya saat pelatihan dilakukan, tetapi kurang baik saat pengujian dilakukan.

C. Convolutional Neural Network (CNN)

CNN merupakan jenis *deep learning* yang memanfaatkan *convolutional layer* sebagai penyusun *neural network* yang dibangun. Banyak penelitian yang membuktikan bahwa CNN lebih baik daripada metode lain dalam melakukan pengolahan teks. Telah dinyatakan bahwa *convolutional layer* pada CNN dapat mengurangi beban komputasi dari *neural network* [9]. Hal ini terjadi karena *convolutional layer* pada dasarnya adalah sebuah *sparse matrix* yang dimensinya lebih kecil dari dimensi data yang diolah.

Teknik konvolusi adalah teknik untuk mengalikan matriks masukan dengan matriks bernama *kernel* untuk menghasilkan keluaran. Teknik konvolusi melakukan perkalian matriks dengan cara yang berbeda dengan perkalian matriks pada umumnya. Pada formula matematis, operasi konvolusi umumnya disimbolkan dengan tanda asterisk (*) [9]. Persamaan (1) menunjukkan formula dasar proses konvolusi.

$$s(x) = (I * K)(x) = \sum_n I(n - x)K(n). \quad (1)$$

Pada (1), I adalah masukan dari proses konvolusi. Pada *text processing*, I merupakan matriks yang merepresentasikan teks yang diolah. Kemudian, K adalah *kernel* dari proses konvolusi yang ukurannya ditentukan sendiri dan s adalah nilai keluaran dari proses konvolusi. Pada bidang *machine learning*, keluaran proses konvolusi biasanya disebut dengan *feature map* [9].

Metode CNN umumnya dimanfaatkan untuk mengolah citra karena *convolution layer* dapat dengan mudah mengekstrak fitur-fitur kecil dari suatu citra. Namun, beberapa penelitian telah menerapkan CNN untuk melakukan pengolahan teks berita [3], [8]. Hasil akurasi yang ditunjukkan oleh kedua penelitian tersebut juga dapat dikatakan cukup baik karena memiliki akurasi $\geq 80\%$.

D. Long Short-Term Memory (LSTM)

LSTM merupakan pengembangan dari metode RNN melalui penambahan sel (*cell*) LSTM di dalam arsitektur RNN. LSTM telah sukses menyelesaikan berbagai permasalahan, seperti *handwriting recognition*, *speech recognition*, *handwriting generation*, dan *image captioning* [9]. LSTM memungkinkan arsitektur *machine learning* menyimpan bobot (*weight*) dari suatu perhitungan lebih lama dari RNN. Hal ini disebabkan LSTM memiliki sel-sel LSTM, yaitu sebuah *node* yang memiliki *self-recurrent*. Hal ini menyebabkan LSTM dapat bekerja lebih baik daripada RNN pada data dengan sekuens yang lebih panjang [9].

Pada LSTM, proses *recurrent* atau perulangan dilakukan pada level *node* dan *layer*. Sel-sel LSTM dikendalikan oleh *input gate* untuk mengingat atau melupakan informasi yang dimiliki berdasarkan keluaran dari LSTM. Referensi [2] mengimplementasikan LSTM untuk melakukan klasifikasi teks. Hasil yang diperoleh juga dapat dikatakan cukup baik dengan nilai akurasi $\geq 90\%$.

E. Convolutional Long Short-Term Memory (C-LSTM)

CNN dan LSTM adalah dua buah metode yang paling umum digunakan untuk menyelesaikan masalah terkait klasifikasi teks [6]. Akan tetapi, CNN yang dimanfaatkan untuk melakukan pengolahan teks sering kali mengabaikan keterkaitan antar konteks di dalam dokumen teks [2], [4]. Hal ini menyebabkan banyak peneliti mengganti metode CNN dengan LSTM. Walaupun LSTM memiliki kelebihan untuk memproses keterkaitan antar konteks di dalam teks, CNN masih lebih unggul daripada LSTM dalam proses ekstraksi fitur atau mengambil fitur di dalam dokumen teks. Oleh karena itu, dikombinasikan CNN dan LSTM untuk memperoleh kelebihan masing-masing metode [4], [7].

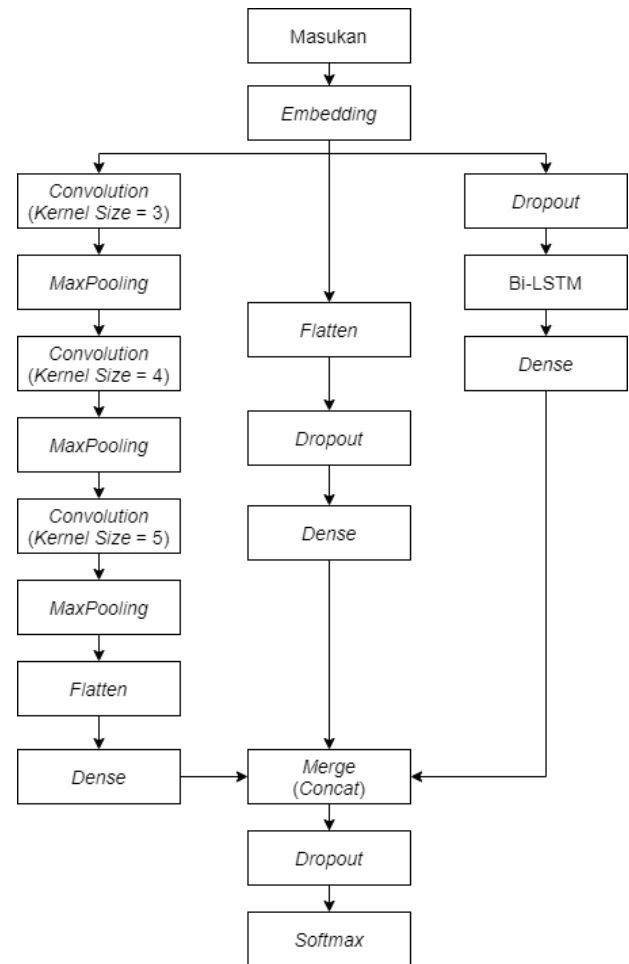
Referensi [4] menggabungkan CNN dan LSTM secara paralel, sehingga dokumen berita yang dimasukkan ke dalam arsitektur *machine learning* diproses secara bersamaan oleh CNN dan LSTM. Hasil dari kedua metode ini kemudian digabungkan untuk memperoleh hasil pengklasifikasian yang dilakukan. Hasil penggabungan kedua metode ini memperlihatkan akurasi yang lebih baik daripada saat kedua metode tersebut dibuat terpisah [4].

Arsitektur C-LSTM yang digunakan pada makalah ini ditunjukkan pada Gbr. 2. Arsitektur yang dibangun adalah hasil konstruksi ulang rancangan yang terdapat pada [4]. Namun, pada makalah ini dilakukan modifikasi untuk mengatasi *overfitting* yang terjadi pada arsitektur C-LSTM.

F. Pengaturan Hyperparameter C-LSTM

Referensi [4] memperlihatkan kerangka arsitektur C-LSTM yang digunakan untuk melakukan klasifikasi teks. Namun, tidak disampaikan nilai *hyperparameter* yang digunakan untuk membangun C-LSTM yang diajukan [4]. Oleh karena itu, dilakukan proses pengaturan *hyperparameter* pada makalah ini. Pengaturan *hyperparameter* dilakukan melalui proses studi pada penelitian sejenis terkait klasifikasi teks, khususnya klasifikasi teks berita.

Hyperparameter juga dapat diperoleh dengan metode *grid search*, yaitu metode untuk mencobakan kombinasi beberapa *hyperparameter*, kemudian membandingkan setiap kombinasi dengan metrik perhitungan yang sama. Akan tetapi, pemilihan nilai kombinasi dari *hyperparameter* yang digunakan pada *grid search* tetap merujuk dari beberapa sumber, seperti penelitian terkait klasifikasi teks dan artikel terkait pengimplementasian *deep learning*. *Grid search* dipilih karena mudah untuk diimplementasikan. *Grid search* juga cukup efektif untuk melakukan *tuning* pada variasi jumlah *hyperparameter* yang tidak terlalu besar [10]. Tabel I memperlihatkan pengaturan *hyperparameter* yang digunakan untuk membangun C-LSTM pada makalah ini.



Gbr. 2 Arsitektur C-LSTM.

IV. HASIL DAN PEMBAHASAN

Setelah membangun metode C-LSTM dan mengatur *hyperparameter* pada metode tersebut, dilakukan sebuah eksperimen untuk mengamati kinerja metode C-LSTM dalam mengklasifikasi berita berbahasa Indonesia. Eksperimen dilakukan pada tiga buah nilai, yaitu jumlah dokumen untuk latihan, nilai *learning rate*, dan ukuran *batch*. Gbr. 3 memperlihatkan variasi masing-masing nilai dan cara nilai tersebut dikombinasikan. Terlihat pada Gbr. 3 bahwa variasi nilai yang telah disebutkan sebelumnya dikombinasikan satu sama lain secara *cross-product*. Hal ini dilakukan untuk memperkaya skenario eksperimen yang dilakukan.

Dokumen yang digunakan untuk melakukan pelatihan dibentuk dari *dataset* yang dimiliki. Pemilihan *subset* dokumen yang digunakan dilakukan secara acak (*random sampling*). Akan tetapi, jumlah dokumen untuk masing-masing kategori tetap diatur dengan komposisi yang sama.

A. Metrik Perhitungan Kinerja C-LSTM

Untuk mempermudah perhitungan kinerja metode C-LSTM, digunakan sebuah metrik perhitungan *F1-score* untuk menilai kemampuan metode C-LSTM melakukan klasifikasi teks bahasa Indonesia. Nilai *F1-score* dihitung dengan memanfaatkan dua buah metrik perhitungan lainnya, yaitu

TABEL I
PENGATURAN *HYPERPARAMETER* YANG DIGUNAKAN

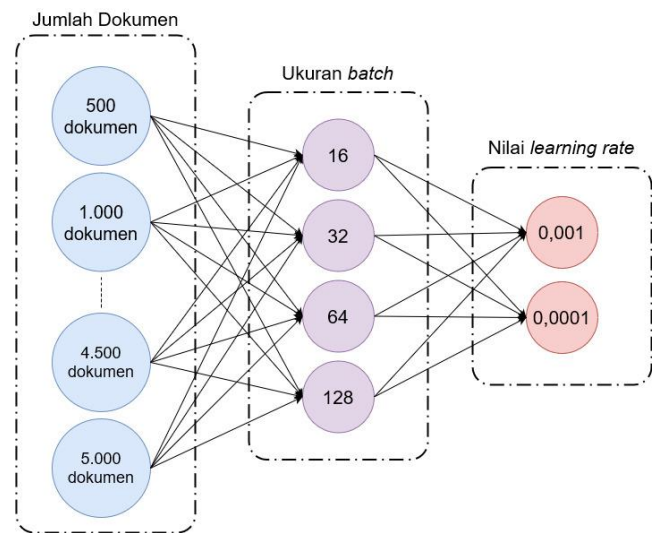
<i>Hyperparameter</i>	Nilai yang Dipilih	Sumber
Jumlah <i>node</i> pada <i>convolutional layer</i>	300	Penelitian yang membahas pemanfaatan CNN untuk klasifikasi teks [3]
Jumlah <i>node</i> pada LSTM dan <i>dense layer</i>	128	<i>Grid search</i> terhadap lima nilai, yaitu 32, 64, 128, 256, dan 512.
<i>Dropout rate</i> pada <i>dropout layer</i>	0,6	<i>Grid search</i> terhadap sembilan nilai yaitu 0,1–0,9. Nilai ini dipilih untuk memeriksa seluruh kemungkinan nilai <i>dropout</i> .
<i>Dropout rate</i> pada LSTM <i>layer</i>	0,9	<i>Grid search</i> terhadap sembilan nilai, yaitu 0,1–0,9. Nilai ini dipilih untuk memeriksa seluruh kemungkinan nilai <i>dropout</i> .
<i>Weight norm</i>	1	<i>Grid search</i> terhadap lima nilai, yaitu 1–5. Nilai ini diperoleh dari hasil membaca artikel mengenai proses mengatasi <i>overfitting</i> pada <i>deep learning</i> .
<i>Regularizer</i>	0,001	<i>Grid search</i> pada empat nilai, yaitu 0,1; 0,01; 0,001; dan 0,0001. Nilai ini diperoleh dari hasil membaca artikel mengenai proses mengatasi <i>overfitting</i> pada <i>deep learning</i> .
<i>Optimizer</i>	Adam	<i>Grid search</i> terhadap beberapa variasi <i>optimizer</i> yang tersedia di <i>library</i> Keras (RMSprop, Adam, SGD, dan Adamax). Namun, rancangan asli dari C-LSTM memanfaatkan Adam sebagai <i>optimizer</i> .
<i>Activation function (hidden layer)</i>	ReLU	<i>Grid search</i> pada variasi <i>activation function</i> yang tersedia di <i>library</i> Keras (ReLU, Softplus, Softsign, TanH, dan Sigmoid). Akan tetapi, beberapa penelitian memanfaatkan ReLU sebagai <i>activation function</i> dari <i>hidden layer</i> pada arsitektur <i>deep learning</i> yang dibangun untuk melakukan <i>text classification</i> [2]–[4], [6].

precision dan *recall*. Persamaan (2) dan (3) memperlihatkan rumus untuk menghitung nilai *precision* dan *recall*.

$$precision = \frac{TP}{TP+FP} \quad (2)$$

$$recall = \frac{TP}{TP+FN} \quad (3)$$

dengan *TP* adalah *true positive*, yang menunjukkan jumlah dokumen berita pada suatu kategori yang berhasil diklasifikasi dengan benar oleh C-LSTM; *FP* adalah *false positive*, menunjukkan jumlah berita dengan kategori tertentu yang dikategorikan dengan kurang tepat; sedangkan *FN* adalah *false negative*, yang menunjukkan jumlah berita yang diklasifikasikan sebagai suatu kategori tertentu, tetapi



Gbr. 3 Kombinasi variabel eksperimen.

TABEL II
PENGARUH JUMLAH DOKUMEN LATIH PADA *F1-SCORE*

Jumlah Dokumen	Nilai <i>F1-Score</i>
500	0,2789
1.000	0,6312
1.500	0,6917
2.000	0,7537
2.500	0,7896
3.000	0,8663
3.500	0,8396
4.000	0,8413
4.500	0,8802
5.000	0,8934

sebenarnya bukan merupakan kategori tersebut. Nilai *precision* dan *recall* dihitung untuk masing-masing kategori, sehingga meski *FP* dan *FN* sama-sama menunjukkan klasifikasi yang salah, tetapi perhitungan untuk kedua nilai tersebut bisa jadi berbeda. Persamaan (4) menunjukkan rumus untuk menghitung *F1-score* yang dijadikan metrik perbandingan pada makalah ini.

$$F1 - score = \frac{2 \times (recall \times precision)}{(recall + precision)} \quad (4)$$

Nilai *F1-score* yang dihitung menggunakan (4) dijadikan metrik perhitungan kinerja C-LSTM dalam mengklasifikasikan berita bahasa Indonesia dengan tiga buah kategori (“Internasional”, “Nasional”, dan “Regional”). Nilai ini dipilih karena mampu memperlihatkan perhitungan yang sensitif terhadap *dataset* yang memiliki *minor class* di dalamnya [11].

B. Pengaruh Jumlah Dokumen Latih pada *F1-Score*

Jumlah dokumen latih adalah salah satu faktor yang berpengaruh terhadap kinerja suatu *machine learning*. Tabel II memperlihatkan pengaruh jumlah dokumen latih terhadap nilai *F1-score* pada masing-masing variasi yang telah ditentukan. Nilai *F1-score* yang terdapat pada Tabel II merupakan nilai rerata *F1-score* untuk seluruh skenario eksperimen yang dilakukan. Seperti yang telah disampaikan sebelumnya, proses eksperimen dilakukan dengan mengombinasikan seluruh

variasi nilai yang terdapat pada Gbr. 2. Oleh karena itu, nilai 0,8934 pada jumlah dokumen 5.000 diperoleh dengan rata-rata hasil *F1-score* untuk jumlah dokumen 5.000 dengan berbagai variasi *learning rate* dan ukuran *batch*.

Terlihat pada Tabel II bahwa nilai *F1-score* mengalami peningkatan seiring dengan bertambahnya jumlah dokumen yang digunakan untuk pelatihan metode C-LSTM. Pada Tabel II juga terlihat sebuah titik jenuh pada jumlah dokumen 3.000. Titik jenuh merupakan titik ketika nilai *F1-score* sudah berhenti mengalami peningkatan yang signifikan. Kendati nilai *F1-score* bisa mencapai 0,8934 atau 89,34% pada jumlah dokumen 5.000, peningkatan yang diperoleh dari jumlah dokumen 3.000 ke 5.000 hanya sebesar 0,0271 (2,71%). Nilai ini tentunya tidak terlalu signifikan jika dibandingkan dengan peningkatan dari jumlah dokumen sebanyak 2.500 ke 3.000.

C. Pengaruh Ukuran Batch pada F1-Score

Ukuran *batch* adalah salah satu *hyperparameter* yang memengaruhi kinerja suatu *neural network*. Hal ini terjadi karena ukuran *batch* secara tidak langsung dapat memengaruhi besarnya bobot yang terdapat dalam arsitektur *neural network* [12], [13]. Pada penelitian yang dilakukan, ukuran *batch* menjadi salah satu nilai yang diamati pengaruhnya terhadap *F1-score* C-LSTM.

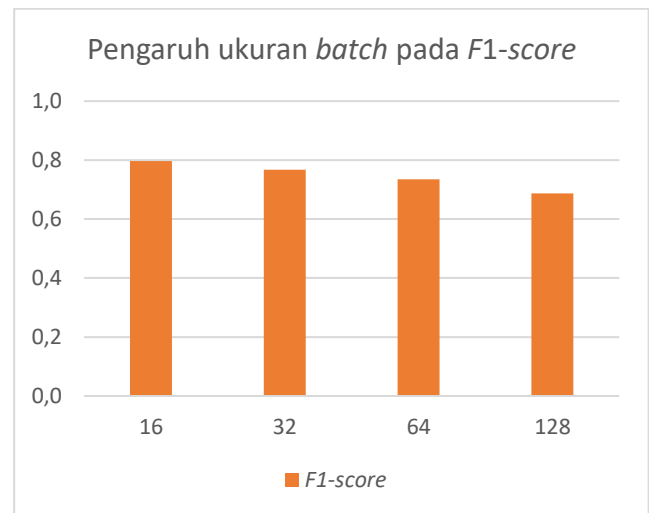
Gbr. 4 memperlihatkan grafik pengaruh ukuran *batch* terhadap nilai *F1-score*. Nilai *F1-score* yang terdapat pada Gbr. 4 diperoleh dengan rata-rata nilai *F1-score* dengan ukuran *batch* yang sama. Hal ini dilakukan untuk melihat secara umum pengaruh ukuran *batch* terhadap *F1-score*. Terlihat dari Gbr. 4, ukuran *batch* 16 menunjukkan rata-rata *F1-score* yang lebih baik daripada ukuran *batch* lainnya, yaitu mendekati 0,8 (80%).

Terlihat adanya keterkaitan secara negatif antara ukuran *batch* dengan *F1-score*. Hal ini dilihat dari menurunnya nilai *F1-score* seiring bertambahnya ukuran *batch*. Hal ini bisa jadi dipengaruhi oleh jumlah *batch* yang terbentuk saat pelatihan dilakukan. Ukuran *batch* yang lebih kecil pasti akan menghasilkan jumlah *batch* yang lebih banyak daripada ukuran *batch* yang besar pada jumlah dokumen latih yang sama. Hal ini juga tentunya berpengaruh pada waktu pelatihan yang dibutuhkan. Jumlah *batch* yang lebih banyak tentunya akan membutuhkan waktu yang lebih lama dalam proses pelatihan C-LSTM.

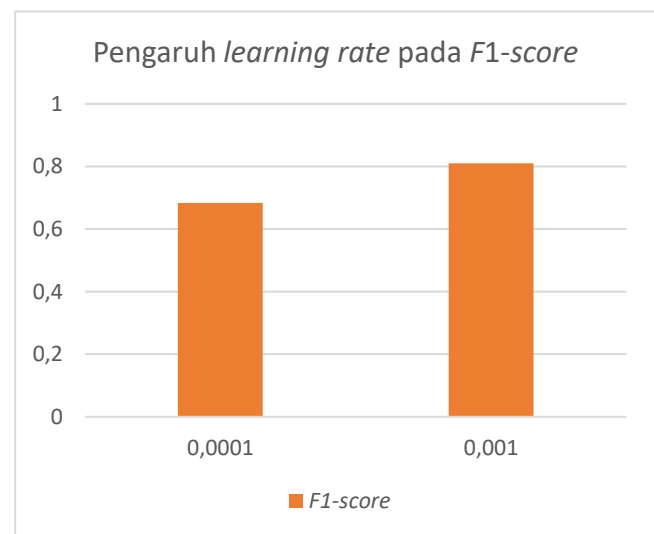
Dengan demikian, pemilihan ukuran *batch* yang lebih kecil seperti 16 atau 32 sangat disarankan untuk meningkatkan nilai *F1-score* dari proses klasifikasi teks berita bahasa Indonesia memanfaatkan C-LSTM. Ukuran *batch* 32 masih dapat dipilih karena dapat menghasilkan *F1-score* yang masih cukup tinggi, tetapi dengan waktu pelatihan yang lebih cepat daripada ukuran *batch* 16.

D. Pengaruh Nilai Learning Rate pada F1-Score

Sama halnya dengan ukuran *batch*, *learning rate* adalah salah satu *hyperparameter* yang dapat memengaruhi bobot yang terdapat dalam arsitektur *neural network*. Pada makalah ini, dipilih dua buah *learning rate*, yaitu 0,001 dan 0,0001. Kedua *learning rate* ini diamati pengaruhnya terhadap nilai *F1-score*. Gbr. 5 memperlihatkan grafik perbandingan nilai *learning rate* terhadap *F1-score* C-LSTM. Seperti yang dapat



Gbr. 4 Grafik pengaruh ukuran *batch* pada *F1-score*.



Gbr. 5 Pengaruh *learning rate* pada *F1-score*.

dilihat, nilai *learning rate* 0,001 memperlihatkan *F1-score* yang lebih tinggi, yaitu 0,8098 (80,98%). Nilai ini lebih baik 0,1264 (12,64%) dari *learning rate* sebesar 0,0001.

Meskipun *learning rate* 0,001 lebih baik daripada 0,0001, *learning rate* tidak boleh diatur terlalu besar ataupun terlalu kecil. Hal ini perlu dilakukan untuk memastikan *machine learning* mempelajari data dengan baik. Besarnya *learning rate* dapat memengaruhi kecepatan *machine learning* dalam melakukan pembelajaran, sehingga nilai *learning rate* 0,001 adalah besaran yang disarankan pada penggunaan C-LSTM dalam makalah ini.

E. Kinerja C-LSTM pada Hyperparameter Pilihan

Berdasarkan eksperimen terhadap berbagai *hyperparameter*, ukuran *batch* 16 menghasilkan nilai *F1-score* yang paling baik di antara ukuran *batch* lainnya, sehingga dalam membangun C-LSTM, ukuran *batch* 16 adalah yang disarankan untuk digunakan. Begitu pula dengan nilai *learning rate* sebesar 0,001, yang merupakan nilai *default* dari *optimizer* Adam yang digunakan saat membangun C-LSTM. Oleh karena itu, pada

TABEL III
F1-SCORE C-LSTM DENGAN HYPERPARAMETER TERPILIH

Jumlah Dokumen	F1-Score
500	0,5748
1.000	0,7302
1.500	0,8077
2.000	0,8725
2.500	0,8920
3.000	0,9187
3.500	0,9153
4.000	0,9202
4.500	0,9203
5.000	0,9327

makalah ini, C-LSTM diatur menggunakan ukuran *batch* sebesar 16 dan *learning rate* 0,001, sebelum dibandingkan nilai *F1-score*-nya dengan metode lain, seperti LSTM dan CNN. Tabel III memperlihatkan nilai *F1-score* untuk C-LSTM yang telah diatur ukuran *batch* dan *learning rate*-nya pada variasi jumlah dokumen berita yang terdapat pada Gbr. 3.

Terlihat pada Tabel III bahwa C-LSTM dengan ukuran *batch* dan *learning rate* terpilih menghasilkan nilai *F1-score* yang lebih baik, dengan *F1-score* mencapai 0,9327 (93,27%). Terlihat juga bahwa titik jenuh pada C-LSTM dengan *hyperparameter* terpilih terdapat pada jumlah dokumen berita sebanyak 2.000 berita. Jumlah dokumen 2.000 dianggap sebagai titik jenuh karena peningkatan yang terjadi setelah titik tersebut tidak lagi signifikan seperti sebelum titik tersebut.

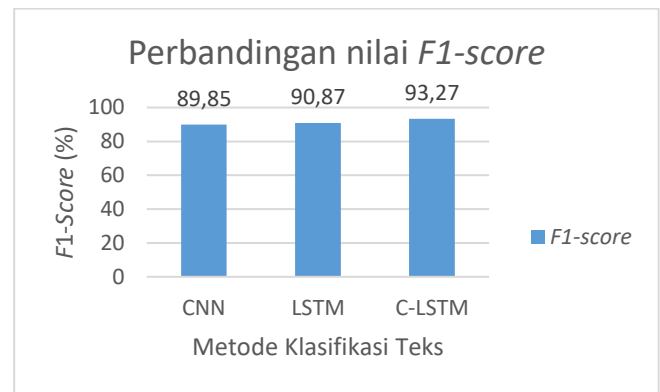
Berdasarkan Tabel III, dapat diambil kesimpulan bahwa jumlah dokumen memang benar berpengaruh terhadap nilai *F1-score*. Namun, jika dibandingkan dengan Tabel II, terlihat bahwa terdapat pergeseran titik jenuh saat *hyperparameter* diatur dengan spesifik. Penggunaan ukuran *batch* 16 dan *learning rate* 0,001 terlihat memberikan dampak positif bagi C-LSTM, karena dengan jumlah dokumen yang sama, nilai *F1-score* yang diperoleh berada di atas rata-rata *F1-score* seperti yang terlihat pada Tabel II.

F. Perbandingan C-LSTM dengan LSTM dan CNN

Subbagian ini secara khusus membahas perbandingan nilai *F1-score* dari C-LSTM, CNN, dan LSTM. Perbandingan dilakukan untuk melihat nilai *F1-score* C-LSTM, lebih baik daripada CNN dan LSTM atau tidak, pada studi kasus berita berbahasa Indonesia. Gbr. 6 memperlihatkan perbandingan CNN, LSTM, dan C-LSTM untuk klasifikasi teks berita bahasa Indonesia. *Dataset* yang digunakan untuk membandingkan C-LSTM, CNN, dan LSTM adalah *dataset* dokumen berita dengan jumlah 5.000 dokumen berita.

Dapat dilihat pada Gbr. 6 bahwa C-LSTM memiliki nilai *F1-score* yang lebih baik daripada CNN maupun LSTM. C-LSTM memiliki nilai *F1-score* 2,4% lebih baik dari LSTM dan 3,42% lebih baik dari CNN. Dari sini dapat dilihat bahwa C-LSTM memiliki kinerja (nilai *F1-score*) yang lebih baik daripada CNN maupun LSTM saat tidak digabungkan.

Tabel IV juga memperlihatkan bahwa C-LSTM cukup sensitif pada kategori "Regional", yang memiliki jumlah dokumen berita lebih sedikit daripada "Nasional" dan



Gbr. 6 Perbandingan *F1-score* CNN, LSTM, dan C-LSTM.

TABEL IV
NILAI F1-SCORE PER KATEGORI UNTUK C-LSTM

Kategori	Precision	Recall	F1-Score	Jumlah Dokumen
Internasional	0,9437	0,9153	0,9293	366
Nasional	0,9398	0,9690	0,9542	548
Regional	0,8889	0,7442	0,8101	86
Rata-rata	0,9368	0,9300	0,9327	

"Internasional". Hal ini dapat dilihat dari nilai *precision*, *recall*, dan *F1-score* untuk kategori "Regional" yang cukup baik. Dari sini dapat dilihat bahwa C-LSTM yang dibangun mampu mengklasifikasikan dokumen berita bahasa Indonesia dengan baik.

V. KESIMPULAN

Dari penelitian yang telah dilakukan, dapat ditarik kesimpulan bahwa dalam mengklasifikasikan berita berbahasa Indonesia terdapat dua buah metode yang umum digunakan, yaitu CNN dan LSTM. Kedua metode ini dapat dikombinasikan untuk membentuk sebuah metode baru dengan nama C-LSTM. Metode C-LSTM terlihat memiliki kinerja yang lebih baik daripada kedua metode pendahulunya (CNN dan LSTM). Hal ini dapat dilihat dari nilai *F1-score* yang melebihi kedua metode lainnya. C-LSTM dapat memperoleh nilai *F1-score* sebesar 0,9327 (93,27%) pada *dataset* berita yang dipilih, lebih baik 2,4% dari LSTM dan lebih baik 3,42% dari CNN.

Pada penelitian ini juga terlihat bahwa C-LSTM yang dibangun cukup sensitif untuk mengklasifikasikan dokumen berita. Hal ini terlihat dari nilai *precision*, *recall*, dan *F1-score* yang cukup tinggi untuk kategori "Regional" yang lebih sedikit daripada kategori "Nasional" dan "Internasional".

Eksperimen untuk melihat *hyperparameter* terbaik juga telah dilakukan. Ukuran *batch* dan *learning rate* terlihat memengaruhi nilai *F1-score* dari pengklasifikasian berita. Ukuran *batch* yang lebih kecil terlihat menghasilkan nilai *F1-score* yang lebih tinggi daripada ukuran *batch* yang besar. Oleh karena itu, pada makalah ini, ukuran *batch* 16 dipilih sebagai *hyperparameter*. Sementara itu, *learning rate* yang dipilih adalah *learning rate* dengan nilai 0,001, yang merupakan nilai *default* dari *optimizer* Adam yang digunakan.

Dari pengujian yang dilakukan, diketahui juga bahwa terdapat titik jenuh pada jumlah dokumen 2.000 untuk C-

LSTM yang dipilih. Maka, dapat ditarik kesimpulan bahwa dengan dokumen berita sebanyak 2.000, C-LSTM sudah dapat memiliki kinerja (nilai *F1-score*) yang cukup baik. Akan tetapi, penambahan jumlah dokumen latih sampai dengan 5.000 masih memberikan peningkatan kinerja (nilai *F1-score*) pada C-LSTM sampai dengan 6,02%.

REFERENSI

- [1] K. Kowsari, K.J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, dan D. Brown, "Text Classification Algorithms: A Survey," *Inf.*, Vol. 10, No. 4, hal. 1–68, 2019.
- [2] J. Li, Y. Xu, dan H. Shi, "Bidirectional LSTM with Hierarchical Attention for Text Classification," *Proc. 2019 IEEE 4th Adv. Inf. Technol. Electron. Autom. Control Conf. (IAEAC 2019)*, 2019, hal. 456–459.
- [3] M.A. Ramdhani, D.S. Maylawati, dan T. Mantoro, "Indonesian News Classification Using Convolutional Neural Network," *Indones. J. Electr. Eng. Comput. Sci.*, Vol. 19, No. 2, hal. 1000–1009, 2020.
- [4] M. Shi, K. Wang, dan C. Li, "A C-LSTM with Word Embedding Model for News Text Classification," *Proc. - 18th IEEE/ACIS Int. Conf. Comput. Inf. Sci. (ICIS 2019)*, 2019, hal. 253–257.
- [5] C. Li, G. Zhan, dan Z. Li, "News Text Classification Based on Improved Bi-LSTM-CNN," *Proc. - 9th Int. Conf. Inf. Technol. Med. Educ. (ITME 2018)*, 2018, hal. 890–893.
- [6] A. Kulkarni dan A. Shivananda, "Deep Learning for NLP," dalam *Natural Language Processing Recipes*, New York, AS: Apress, 2019, Ch. 6, hal. 185–227.
- [7] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, dan B. Xu, "Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling," *Proc. COLING 2016 - 26th Int. Conf. Comput. Linguist. Tech. Pap.*, 2016, hal. 3485–3495.
- [8] F. Zhu, X. Dong, R. Song, Y. Hong, dan Q. Zhu, "A Multiple Learning Model Based Voting System for News Headline Classification," dalam *Natural Language Processing and Chinese Computing (NLPCC 2017), Lecture Notes in Computer Science*, Vol. 10619, X. Huang, J. Jiang, D. Zhao, Y. Feng, dan Yu Hong, Eds., Cham, Swiss: Springer, 2018, hal. 797–806.
- [9] I. Goodfellow, Y. Bengio, dan A. Courville, *Deep Learning*, Cambridge, AS: The MIT Press, 2016.
- [10] L. Yang dan A. Shami, "On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice," *Neurocomput.*, Vol. 415, hal. 295–316, 2020.
- [11] Suyanto, "Ukuran Evaluasi Model Klasifikasi," dalam *Machine Learning Tingkat Dasar dan Lanjut*, Bandung, Indonesia: INFORMATIKA, 2018, hal. 334.
- [12] I. Kandel dan M. Castelli, "The Effect of Batch Size on the Generalizability of the Convolutional Neural Networks on a Histopathology Dataset," *ICT Express*, Vol. 6, No. 4, hal. 312–315, 2020.
- [13] P.M. Radiuk, "Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets," *Inf. Technol. Manag. Sci.*, Vol. 20, No. 1, hal. 20–24, 2018.