HOAQ: JURNAL TEKNOLOGI INFORMASI, Volume 17 Nomor 2, Desember 2026, p241-251
p-ISSN: 2337-5280, e-ISSN: 26207427
DOI: https://doi.org/10.52972/hoaq.vol17no2

EVALUASI *MAXIMUM SEQUENCE LENGTH* PADA INDOBERT-BILSTM-CRF UNTUK *NAMED ENTITY RECOGNITION* BAHASA INDONESIA

Calvin Chang

Program Studi Informatika, Universitas Widya Dharma Pontianak
Jl. HOS Cokroaminoto, Pontianak - Kalimantan Barat, Indonesia
Email: calvin_chang@widyadharma.ac.id

ABSTRAK

Maximum sequence length merupakan salah satu parameter penting pada model berbasis *Transformer* karena memengaruhi cakupan input sekaligus kebutuhan komputasi. Penelitian ini menganalisis pengaruh *maximum sequence length* terhadap kinerja dan efisiensi komputasi model IndoBERT-BiLSTM-CRF pada tugas *Named Entity Recognition* Bahasa Indonesia menggunakan dataset IDNer-News-2K. Empat konfigurasi diuji, yaitu 64, 128, 256, dan 512 token, dengan parameter model lainnya dipertahankan tetap. Setiap konfigurasi dijalankan menggunakan tiga *random seed*. Kinerja dievaluasi menggunakan Token Macro-F1 dan Entity F1, sedangkan efisiensi dianalisis berdasarkan training time, penggunaan GPU *memory, training throughput*, dan *inference time*. Hasil menunjukkan bahwa MAX_LEN=64 masih menyebabkan *truncation* sekitar 0,55% dan menghasilkan *Entity* F1 sebesar 0,9119. MAX_LEN=128 menghilangkan seluruh *truncation* serta mencapai 100% *word coverage* dan *token coverage*, dengan *Entity* F1 sebesar 0,9171. Nilai Entity F1 tertinggi diperoleh pada MAX_LEN=256 sebesar 0,9177, tetapi peningkatannya terhadap MAX_LEN=128 sangat kecil dan disertai kenaikan biaya komputasi. Pada MAX_LEN=512, waktu pelatihan dan penggunaan GPU *memory* meningkat secara substansial, sementara *Entity* F1 menurun menjadi 0,9159. Temuan ini menunjukkan bahwa peningkatan *maximum sequence length* setelah seluruh input tercakup memberikan diminishing return. Berdasarkan keseimbangan antara kinerja dan efisiensi, MAX_LEN=128 merupakan konfigurasi yang paling rasional untuk dataset yang digunakan.

Kata kunci: *Named Entity Recognition*, IndoBERT, BiLSTM-CRF, *maximum sequence length*, efisiensi komputasi

ABSTRACT

Maximum sequence length is an important parameter in Transformer-based models because it affects both input coverage and computational requirements. This study analyzes the effect of maximum sequence length on the performance and computational efficiency of an IndoBERT-BiLSTM-CRF model for Indonesian Named Entity Recognition using the IDNer-News-2K dataset. Four configurations were evaluated, namely 64, 128, 256, and 512 tokens, while keeping the remaining model parameters unchanged. Each configuration was run using three random seeds. Model performance was evaluated using Token Macro-F1 and Entity F1, while computational efficiency was assessed based on training time, GPU memory usage, training throughput, and inference time. The results show that MAX_LEN=64 still caused approximately 0.55% truncation and achieved an Entity F1 of 0.9119. MAX_LEN=128 eliminated truncation completely and achieved 100% word coverage and token coverage, with an Entity F1 of 0.9171. The highest Entity F1 was obtained at MAX_LEN=256, reaching 0.9177, but the improvement over MAX_LEN=128 was marginal and accompanied by higher computational cost. At MAX_LEN=512, training time and GPU memory usage increased substantially, while Entity F1 decreased to 0.9159. These findings indicate that increasing maximum sequence length after full input coverage has been achieved results in diminishing returns. Considering the trade-off between performance and computational efficiency, MAX_LEN=128 provides the most practical configuration for the dataset used in this study.

Keywords: Named Entity Recognition, IndoBERT, BiLSTM-CRF, maximum sequence length, computational efficiency

1. PENDAHULUAN

Named Entity Recognition (NER) merupakan salah satu tugas utama dalam *Natural Language Processing* (NLP) yang berfokus pada pengenalan dan pengelompokan entitas dalam teks ke dalam kategori tertentu, seperti *person, organization*, dan *location*. Kemunculan *pre-trained language* model berbasis *Transformer* telah secara signifikan meningkatkan performa NER melalui pembentukan representasi kontekstual yang lebih kaya serta kemampuan mengatasi ambiguitas kata berdasarkan konteks.

Dalam pemrosesan Bahasa Indonesia, IndoBERT telah banyak dimanfaatkan untuk berbagai tugas ekstraksi informasi, termasuk NER. [1] menunjukkan bahwa kombinasi IndoBERT dengan model sekuensial dan CRF dapat

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

dimanfaatkan untuk ekstraksi entitas finansial dalam Bahasa Indonesia, sedangkan [2] menerapkan pendekatan hibrida berbasis Transformer dan BiLSTM pada korpus berita daring berbahasa Indonesia. Studi lain juga melaporkan bahwa IndoBERT efektif untuk tugas NER Bahasa Indonesia dengan memanfaatkan representasi kontekstual dan pembelajaran transfer [3], [4].

Meskipun model berbasis *Transformer* memiliki kemampuan representasi kontekstual yang kuat, kinerjanya tetap dipengaruhi oleh konfigurasi input, salah satunya maximum sequence length. Parameter ini menentukan jumlah maksimum token yang dapat diproses dalam satu *sequence*. Jika panjang input melampaui nilai tersebut, sebagian token harus dipotong atau diproses secara terpisah, sehingga berpotensi menimbulkan hilangnya informasi kontekstual maupun entitas yang berada di bagian akhir *sequence.* [5] menunjukkan bahwa batas panjang input BERT merupakan aspek penting dalam pemrosesan teks biomedis yang panjang, sehingga diperlukan strategi pemisahan atau pengaturan panjang input untuk mempertahankan informasi sebanyak mungkin. Dalam aplikasi NER, pemilihan panjang *sequence* yang sesuai juga krusial karena model harus mempertahankan konteks yang memadai untuk menentukan batas dan tipe entitas secara akurat [6], [7].

Pemilihan maximum sequence length pada Transformer melibatkan kompromi antara cakupan informasi dan biaya komputasi. *Sequence* yang terlalu pendek meningkatkan risiko *truncation*, sedangkan *sequence* yang terlalu panjang membuat model memproses lebih banyak posisi meskipun banyak *sequence* sebenarnya lebih pendek. Mekanisme *self-attention* pada *Transformer* memiliki kebutuhan komputasi dan memori yang meningkat seiring pertambahan panjang input. [8] menunjukkan bahwa *self-attention* standar memiliki kompleksitas komputasi dan memori yang berskala kuadratik terhadap panjang *sequence*. [9] dalam tinjauannya mengenai *efficient Transformer*, juga menempatkan panjang *sequence* sebagai salah satu faktor utama yang mendorong pengembangan arsitektur Transformer yang lebih efisien. Penelitian terbaru mengenai CAST turut mengindikasikan bahwa peningkatan panjang input dapat menaikkan kebutuhan waktu komputasi dan memori secara signifikan [10].

Dalam penelitian NER berbasis BERT, *maximum sequence length* umumnya ditetapkan sebagai bagian dari konfigurasi eksperimen. [6] menggunakan konfigurasi panjang input tertentu dalam pengembangan jaringan NER berbasis BERT, sedangkan [7] menerapkan BERT untuk de-identifikasi data klinis dengan batas panjang input yang disesuaikan dengan karakteristik data. Studi lain pada domain penerbangan juga menunjukkan pemanfaatan BERT untuk NER dengan pengaturan input dan *fine-tuning* yang diadaptasi terhadap karakteristik domain [11]. Temuan tersebut menegaskan bahwa panjang *sequence* merupakan komponen penting dalam konfigurasi model, meskipun pada banyak penelitian parameter ini lebih sering diperlakukan sebagai nilai tetap daripada dikaji secara khusus sebagai faktor yang memengaruhi performa dan efisiensi model.

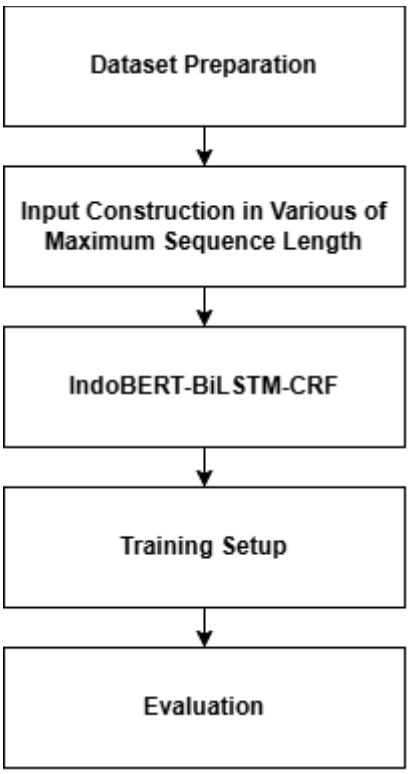
Permasalahan ini juga relevan bagi arsitektur IndoBERT-BiLSTM-CRF. Peningkatan *maximum sequence length* memungkinkan model mempertahankan konteks yang lebih luas, namun sekaligus menambah jumlah token yang harus diproses oleh IndoBERT dan diteruskan ke BiLSTM serta CRF. Jika sebagian besar *sequence* telah terakomodasi secara memadai pada panjang tertentu, peningkatan panjang input lebih lanjut belum tentu memberikan peningkatan kinerja yang sebanding dengan tambahan waktu pelatihan, kebutuhan GPU *memory* yang lebih besar, penurunan *throughput*, serta perpanjangan waktu inferensi. Dengan demikian, pemilihan *maximum sequence length* tidak hanya berdampak pada akurasi model, tetapi juga pada efisiensi pemanfaatan sumber daya komputasi.

Berdasarkan permasalahan tersebut, penelitian ini menganalisis pengaruh *maximum sequence length* terhadap kinerja dan efisiensi komputasi model IndoBERT-BiLSTM-CRF pada NER Bahasa Indonesia. Empat konfigurasi, yaitu 64, 128, 256, dan 512 token, diuji dengan mempertahankan parameter model lainnya tetap. Keempat nilai tersebut dipilih untuk merepresentasikan variasi kapasitas input secara bertingkat dengan peningkatan panjang *sequence* sebesar dua kali lipat, mulai dari konfigurasi input pendek hingga batas maksimum 512 token pada arsitektur BERT standar. Pendekatan ini memungkinkan evaluasi perubahan cakupan informasi dan kebutuhan komputasi pada berbagai tingkat panjang input secara sistematis. Kinerja model dievaluasi menggunakan Token Macro-F1 dan *Entity* F1, sedangkan dampak terhadap data input dianalisis berdasarkan tingkat truncation dan word *coverage*. Selain itu, efisiensi komputasi dianalisis menggunakan waktu pelatihan, penggunaan GPU *memory*, *training throughput*, dan waktu inferensi. Penelitian ini bertujuan mengidentifikasi panjang *sequence* yang memberikan keseimbangan antara cakupan informasi, kinerja NER, dan biaya komputasi, sehingga konfigurasi yang dipilih tidak hanya menghasilkan performa tinggi tetapi juga efisien dalam penggunaan sumber daya.

2. METODE PENELITIAN

Penelitian ini menerapkan pendekatan eksperimental untuk mengkaji dampak variasi *maximum sequence length* terhadap kinerja serta efisiensi komputasi model IndoBERT-BiLSTM-CRF pada tugas *Named Entity Recognition* (NER) berbahasa Indonesia. Dalam penelitian ini diuji empat nilai *maximum sequence length*, yaitu 64, 128, 256, dan 512 token. Seluruh parameter model lainnya dijaga konstan sehingga perubahan kinerja maupun kebutuhan komputasi dapat dikaitkan secara langsung dengan perbedaan panjang *sequence*. Rangkaian tahapan penelitian mencakup persiapan *dataset*, penyusunan input, pemodelan menggunakan IndoBERT-BiLSTM-CRF, proses pelatihan, evaluasi, serta analisis *trade-off* antara kinerja dan efisiensi komputasi.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi



Gambar 1. Kerangka Kerja Metodologi Penelitian

Dataset Preparation

Dataset yang digunakan dalam penelitian ini adalah IDNer-News-2K, yaitu *dataset Named Entity Recognition* Bahasa Indonesia yang berasal dari teks berita daring. *Dataset* ini dikembangkan kembali oleh [12] melalui proses re-annotasi terhadap dataset NER Bahasa Indonesia yang sebelumnya disediakan oleh [12]. Dataset IDNer-News-2K terdiri atas sekitar dua ribu kalimat berita Bahasa Indonesia dan disediakan dalam tiga subset, *yaitu training, development*, dan *test*.

Penelitian ini tidak melibatkan anotator baru karena menggunakan label entitas yang telah tersedia pada dataset IDNer-News-2K. Oleh karena itu, proses pemberian label dan konsistensi anotasi mengacu pada dokumentasi *dataset* yang dipublikasikan oleh pengembangnya. Penelitian ini tidak melakukan pengukuran ulang terhadap kesepakatan antaranotator. Label yang tersedia pada dataset digunakan sebagai anotasi referensi untuk proses pelatihan dan evaluasi model IndoBERT-BiLSTM-CRF.

Dataset menggunakan format CoNLL, di mana setiap baris memuat token, *Part-of-Speech* (POS) *tag*, dan *label Named Entity Recognition* (NER). Label entitas mengikuti skema IOB/BIO dengan tiga kategori utama, yaitu PER, ORG, dan LOC, sedangkan label O digunakan untuk token yang bukan bagian dari entitas. *Dataset* terdiri atas tiga *subset*, yaitu *train, validation*, dan *test*, yang digunakan secara langsung tanpa pembagian ulang. Berdasarkan hasil analisis, keseluruhan dataset terdiri atas 2.340 kalimat dan 48.699 token, dengan rincian 1.464 kalimat pada *train*, 367 pada *validation*, dan 509 pada *test*. Total entitas yang ditemukan sebanyak 3.777, terdiri atas 1.632 PER (43,21%), 1.311 ORG (34,71%), dan 834 LOC (22,08%), dengan rata-rata panjang kalimat keseluruhan sebesar 20,81 token. Tabel 1 menunjukkan statistik *dataset*.

Training set digunakan untuk memperbarui parameter model, *development/validation set* digunakan untuk mengevaluasi model pada setiap *epoch* dan menentukan *checkpoint* terbaik, sedangkan *test set* hanya digunakan untuk evaluasi akhir. Pemisahan tersebut bertujuan menjaga independensi data pengujian selama proses pemilihan model. Sebelum proses pelatihan, *dataset* diperiksa untuk memastikan kesesuaian antara token dan label, kemudian diproses menggunakan tokenizer IndoBERT sesuai variasi maximum sequence length yang digunakan dalam eksperimen.

Tabel 1. Statistik *dataset* IDNer-News-2K

File	Kalimat	Token	PER	ORG	LOC	Total Entitas
Train	1.464	30.248	896	721	579	2.196
Dev	367	7.863	221	189	152	562
Test	509	10.588	515	401	103	1.019

Input Construction in Various of Maximum Sequence Length

Setiap *sequence* dikenai proses tokenisasi menggunakan *tokenizer* IndoBERT. IndoBERT sendiri adalah model bahasa pra-latih yang secara khusus dikembangkan untuk Bahasa Indonesia dan mampu membangun

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

representasi kontekstual yang selaras dengan karakteristik bahasa tersebut [13]. Setelah proses tokenisasi, token khusus seperti [CLS] dan [SEP] disisipkan, kemudian panjang sequence disesuaikan dengan nilai *maximum sequence length* yang diterapkan.

Pemilihan *maximum sequence length* dilakukan menggunakan empat konfigurasi, yaitu 64, 128, 256, dan 512 token. Keempat nilai tersebut digunakan untuk mengevaluasi perubahan kapasitas input secara bertingkat dengan peningkatan panjang sequence sebesar dua kali lipat. MAX_LEN=64 merepresentasikan konfigurasi input yang relatif pendek, sedangkan MAX_LEN=128 dan 256 digunakan untuk mengevaluasi pengaruh perluasan kapasitas input terhadap kinerja dan efisiensi komputasi. Sementara itu, MAX_LEN=512 digunakan sebagai batas maksimum panjang input pada arsitektur BERT standar. Pendekatan ini memungkinkan pengamatan terhadap perubahan *truncation, coverage*, kinerja NER, dan kebutuhan komputasi pada beberapa tingkat panjang input. Pemilihan nilai tersebut merupakan rancangan eksperimen bertingkat dan tidak dimaksudkan sebagai pencarian menyeluruh terhadap seluruh kemungkinan nilai *maximum sequence length*.

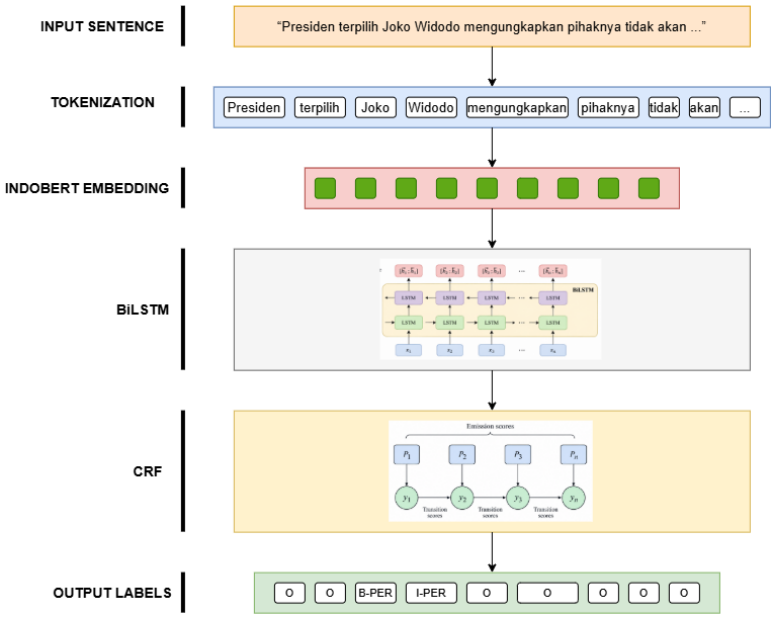
Sequence yang lebih panjang dari nilai MAX_LEN akan mengalami truncation, sedangkan sequence yang lebih pendek akan diberi padding. Untuk mengetahui dampak pembatasan tersebut terhadap informasi input, dilakukan pengukuran *truncation rate* dan *word coverage*. Kompleksitas mekanisme *self-attention* pada arsitektur *Transformer* meningkat seiring dengan bertambahnya panjang *sequence*, sehingga penggunaan input yang lebih panjang dapat mengakibatkan peningkatan kebutuhan sumber daya komputasi dan memori [9].

IndoBERT-BiLSTM-CRF

Model yang digunakan mengintegrasikan IndoBERT, BiLSTM, dan CRF. IndoBERT berperan sebagai *contextual encoder* yang menghasilkan representasi vektor untuk setiap token. Representasi ini selanjutnya diteruskan ke BiLSTM guna menangkap dependensi sekuensial dari arah maju dan mundur. Pemanfaatan BiLSTM memungkinkan informasi sebelum dan sesudah suatu token digunakan secara simultan sehingga arsitektur ini sesuai untuk tugas pelabelan sekuensial seperti NER [14].

Keluaran BiLSTM kemudian diproyeksikan melalui sebuah lapisan linear untuk memperoleh emission score bagi setiap label. Pada tahap akhir, CRF digunakan untuk menentukan kombinasi urutan label dengan skor tertinggi dengan mempertimbangkan ketergantungan antarlabel. Penerapan CRF membantu model menghasilkan urutan label yang lebih konsisten dibandingkan pendekatan yang mengklasifikasikan setiap token secara independen [15].

Gambar 2 merupakan arsitektur IndoBERT-BiLSTM-CRF yang digunkan. Arsitektur yang digunakan terdiri atas IndoBERT sebagai contextual encoder, BiLSTM sebagai pemodel dependensi sekuensial, linear layer sebagai pembentuk emission score, dan CRF sebagai decoder akhir. Misalkan sebuah kalimat direpresentasikan sebagai sequence input $X = [x_1, x_2, \ldots, x_n]$. Sebagai contoh, salah satu kalimat pada dataset adalah "Presiden terpilih Joko Widodo mengungkapkan pihaknya tidak akan ...", dengan Joko berlabel B-PER dan Widodo berlabel I-PER. Sequence tersebut terlebih dahulu diproses oleh tokenizer IndoBERT menjadi token atau *subword*, kemudian ditambahkan token khusus [CLS] dan [SEP] serta dibentuk input IDs dan attention mask. Jika suatu kata terpecah menjadi beberapa *subword*, penyelarasan label dilakukan pada subword pertama sehingga prediksi akhir tetap dikembalikan pada level kata.



Gambar 2. Arsitektur IndoBERT-BiLSTM-CRF

Input hasil tokenisasi selanjutnya diteruskan ke IndoBERT. Untuk setiap posisi token, IndoBERT menghasilkan representasi kontekstual berdimensi tinggi yang dapat dituliskan sebagai: *H=IndoBERT(X) = [h1,*

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

$h_2, \ldots, h_n]$. Berbeda dari representasi kata yang bersifat statis, vektor hih_i merepresentasikan setiap token dengan mempertimbangkan konteks kalimat secara menyeluruh. Pada ilustrasi sebelumnya, representasi token Joko dan Widodo tidak hanya memuat informasi leksikal kedua kata tersebut, tetapi juga mengenkapsulasi konteks di sekitarnya, seperti Presiden, terpilih, dan mengungkapkan. Representasi keluaran IndoBERT kemudian dilewatkan ke lapisan dropout dan diteruskan ke BiLSTM. BiLSTM memproses sekuens dalam dua arah, yakni dari kiri ke kanan dan dari kanan ke kiri, sehingga representasi setiap token merupakan agregasi informasi dari kedua arah tersebut.

$$h_i^{BiLSTM} = [\overrightarrow{h_i}; \overleftarrow{h_i}]$$

Dengan demikian, ketika memproses token Joko, model tidak hanya mempertimbangkan token sebelumnya seperti Presiden terpilih, tetapi juga token sesudahnya seperti Widodo mengungkapkan. Informasi dua arah tersebut membantu model menentukan bahwa Joko Widodo merupakan satu kesatuan entitas bertipe Person.

Keluaran BiLSTM selanjutnya diteruskan ke *linear layer* untuk menghasilkan *emission score* bagi seluruh label BIO pada setiap token. Misalnya, untuk token Joko, model menghasilkan skor terhadap label B-PER, I-PER, B-ORG, I-ORG, B-LOC, I-LOC, dan O. Token Widodo menghasilkan kumpulan skor yang sama. Apabila hanya menggunakan klasifikasi independen, label setiap token dapat dipilih berdasarkan skor tertinggi tanpa mempertimbangkan hubungan dengan token lain. Oleh karena itu, emission score tersebut diteruskan ke lapisan *Conditional Random Field* (CRF) yang mempertimbangkan sekaligus skor token dan skor transisi antarlabel. Secara umum, skor suatu urutan label $Y = (y_1, y_2, \ldots, y_n)$ terhadap masukan X dapat dirumuskan sebagai:

$$S(X, Y) = \sum_{i=1}^{n} P_{i,y_i} + \sum_{i=1}^{n-1} A_{y_i,y_{i+1}}$$

dengan P_{i,y_i} merupakan emission score untuk label y_i pada token ke-i, sedangkan $(A_{y_i,y_{i+1}})$ merupakan skor transisi dari satu label ke label berikutnya. CRF kemudian memilih sequence label dengan skor keseluruhan terbesar. Pada contoh Joko Widodo, kombinasi B-PER → I-PER memperoleh keuntungan karena merupakan transisi yang sesuai dengan struktur BIO.

Training Setup

Konfigurasi pengujian dapat dilihat di Tabel 2. Setiap konfigurasi *maximum sequence length* dilatih menggunakan arsitektur dan hiperparameter yang sama agar perbedaan hasil terutama mencerminkan pengaruh panjang *sequence*. Nilai *maximum sequence length* divariasikan menjadi 64, 128, 256, dan 512, sedangkan *learning rate, batch size*, jumlah *epoch*, ukuran hidden BiLSTM, jumlah *layer, dropout, optimizer*, dan *warmup ratio* dipertahankan tetap. Optimasi parameter dilakukan menggunakan AdamW dengan *learning rate* 5×10^{-5}. Setiap konfigurasi dijalankan sebanyak tiga kali menggunakan random seed 42, 52, dan 62 untuk mengevaluasi kestabilan hasil. Pada setiap *epoch*, model dievaluasi menggunakan validation set, kemudian checkpoint dengan *Validation Entity* F1 tertinggi dipilih untuk evaluasi akhir pada *test set*.

Pada setiap *epoch*, model dievaluasi menggunakan *validation set*. *Checkpoint* dengan nilai *Validation Entity* F1 terbaik disimpan dan kemudian digunakan untuk evaluasi akhir menggunakan test set. Setiap konfigurasi juga dijalankan menggunakan beberapa random seed untuk mengurangi ketergantungan hasil terhadap variasi inisialisasi parameter.

Tabel 2. Konfigurasi Parameter Pengujian

Parameter	Nilai
Arsitektur Model	IndoBERT-BiLSTM-CRF
Pretrained Model	IndoBERT
Tokenizer	IndoBERT Fast Tokenizer
Maximum Sequence Length	64, 128, 256, 512
Learning Rate	5×10^{-5}
Batch Size	8
Epoch	15
Hiden Size	256
Layers	2
Learning Rate Scheduler	*Warmup*
Warmup Ratio	0,1 (10%)
Gradient Cliping	1,0
CRF-Implementation	Pytorch CRF
CRF Loss	*Negative Log-Likelihood*
GPU	T4 Tesla
Dropout	0,1

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Parameter	Nilai
Optimizer	AdamW
Random Seed	42,52,62

Evaluasi

Evaluasi dilakukan terhadap dua aspek utama, yaitu kinerja NER dan efisiensi komputasi. Kinerja model diukur menggunakan Token Macro-F1 dan *Entity* F1-Score. *Entity* F1-Score digunakan untuk mengevaluasi kesesuaian tipe dan span entitas antara hasil prediksi dan anotasi referensi. Efisiensi komputasi dianalisis menggunakan empat ukuran, yaitu: *Training Time, Peak GPU Memory, Training Throughput*, dan *Inference Time*. Selain itu, proses konvergensi dianalisis melalui perkembangan *Validation Entity* F1 pada setiap epoch.

3. HASIL DAN PEMBAHASAN

Hasil Pengujian pada Berbagai *Maximum Sequence Length*

Eksperimen dilakukan dengan menerapkan empat variasi nilai *maximum sequence length*, yaitu 64, 128, 256, dan 512. Setiap konfigurasi dijalankan menggunakan tiga *random seed*, sementara konfigurasi model dan hiperparameter lainnya dijaga agar tetap konstan. Evaluasi kinerja dilakukan melalui penggunaan metrik Token *Macro*-F1-*Score* dan *Entity* F1-Score. Selain itu, perhitungan *precision* dan *recall* pada level entitas dilakukan untuk memberikan gambaran yang lebih mendalam mengenai kemampuan model dalam mengenali entitas. Hasil pengujian ditunjukkan di Tabel 3.

Tabel 3. Kinerja Model Pada Berbagai *Maximum Sequence Length*

Max Len	*Token Macro F1*	*Entity Precision*	*Entity Recall*	*Entity F1*
64	0,9184 ± 0,0043	0,9071 ± 0,0010	0,9169 ± 0,0170	0,9119 ± 0,0079
128	0,9260 ± 0,0073	0,9148 ± 0,0070	0,9195 ± 0,0080	0,9171 ± 0,0063
256	0,9252 ± 0,0065	0,9130 ± 0,0084	0,9225 ± 0,0035	0,9177 ± 0,0051
512	0,9180 ± 0,0064	0,9140 ± 0,0019	0,9179 ± 0,0034	0,9159 ± 0,0024

Hasil pada Tabel 3 menunjukkan bahwa peningkatan maximum sequence length dari 64 menjadi 128 diikuti kenaikan Token *Macro*-F1 dari 0,9184 menjadi 0,9260 dan *Entity* F1 dari 0,9119 menjadi 0,9171. Peningkatan MAX_LEN dari 128 menjadi 256 hanya menghasilkan kenaikan *Entity* F1 sebesar 0,0006 atau sekitar 0,06 poin persentase, sedangkan Token Macro-F1 mengalami sedikit penurunan dari 0,9260 menjadi 0,9252. Pada MAX_LEN=512, Entity F1 menurun menjadi 0,9159 dan Token Macro-F1 menjadi 0,9180. Hasil tersebut menunjukkan bahwa peningkatan maximum sequence length tidak selalu diikuti peningkatan kinerja NER secara konsisten.

Perbedaan kinerja tersebut dapat dikaitkan dengan cakupan input dan karakteristik panjang *sequence* pada dataset. Pada MAX_LEN=64, sebagian sequence pada data latih mengalami *truncation* sehingga terdapat token yang tidak dapat diproses oleh model. Peningkatan MAX_LEN menjadi 128 memungkinkan seluruh *sequence* pada data latih dipertahankan tanpa *truncation*. Kondisi ini berpotensi memberikan informasi kontekstual yang lebih lengkap untuk membantu model mengenali batas dan jenis entitas. Namun, peningkatan MAX_LEN menjadi 256 dan 512 tidak lagi menambah cakupan informasi pada data latih karena seluruh *sequence* telah tercakup pada MAX_LEN=128. Pada konfigurasi yang menggunakan *fixed padding*, peningkatan panjang input justru memperbesar jumlah posisi *padding* dan kebutuhan komputasi tanpa menambahkan informasi linguistik baru. Selain itu, perbedaan performa dapat dipengaruhi oleh variasi proses optimisasi dan penanganan padding dalam arsitektur model. Hal ini terlihat dari hasil pengujian menggunakan tiga random seed, di mana MAX_LEN=256 menghasilkan *Entity* F1 lebih tinggi pada seed 42, sedangkan MAX_LEN=128 menghasilkan *Entity* F1 lebih tinggi pada seed 52 dan 62. Dengan demikian, peningkatan panjang *sequence* setelah cakupan input terpenuhi tidak menjamin peningkatan kinerja NER, sementara kebutuhan komputasi terus bertambah.

Selisih nilai *Entity* F1 antara konfigurasi MAX_LEN=128 dan MAX_LEN=256 menunjukkan margin yang lebih kecil dibandingkan dengan standar deviasi pada masing-masing konfigurasi tersebut. Hasil tersebut menunjukkan bahwa peningkatan performa paling signifikan terjadi saat panjang sekuens ditingkatkan dari 64 menjadi 128. Setelah mencapai ambang batas tersebut, performa mulai menunjukkan pola saturasi.

Tabel 4. Hasil Pengujian Berdasarkan *Maximum Sequence Length* dan *Random Seed*

Max Len	*Seed*	*Token Macro F1*	*Entity Precision*	*Entity Recall*	*Entity F1*
64	42	0,9202	0,9063	0,9241	0,9151
64	52	0,9215	0,9067	0,9291	0,9178
64	62	0,9135	0,9083	0,8975	0,9029
128	42	0,9228	0,9073	0,9127	0,9100
128	52	0,9208	0,9212	0,9176	0,9194
128	62	0,9343	0,9158	0,9284	0,9220

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

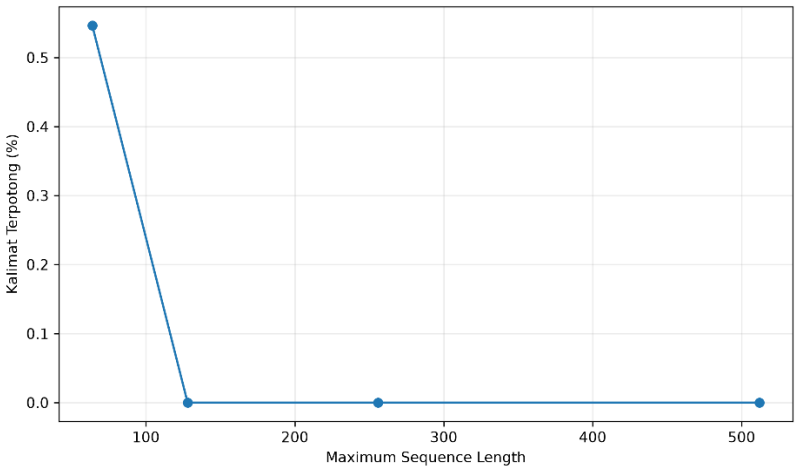
Max Len	*Seed*	*Token Macro F1*	*Entity Precision*	*Entity Recall*	*Entity F1*
256	42	0,9326	0,9225	0,9235	0,9230
256	52	0,9205	0,9094	0,9254	0,9173
256	62	0,9225	0,9070	0,9185	0,9127
512	42	0,9234	0,9158	0,9176	0,9167
512	52	0,9197	0,9143	0,9215	0,9179
512	62	0,9110	0,9119	0,9146	0,9133

Berdasarkan hasil di Tabel 4, variasi *maximum sequence length* menunjukkan perbedaan kinerja antar-*random seed*. Pada MAX_LEN=64, *Entity* F1 berkisar antara 0,9029 hingga 0,9178. Peningkatan MAX_LEN menjadi 128, dengan nilai tertinggi sebesar 0,9220 pada seed 62. Sementara itu, MAX_LEN=256 memperoleh hasil individual tertinggi mencapai 0,9230 pada seed 42. Pada MAX_LEN=512.

Perbandingan hasil antar-*random seed* menunjukkan bahwa konfigurasi dengan *Entity* F1 tertinggi tidak selalu sama. MAX_LEN=256 menghasilkan nilai tertinggi pada *seed* 42, sedangkan MAX_LEN=128 menghasilkan nilai tertinggi pada seed 52 dan 62. Perbedaan rata-rata *Entity* F1 antara MAX_LEN=128 dan 256 hanya sebesar 0,0006. Hal ini menunjukkan bahwa keunggulan rata-rata MAX_LEN=256 tidak terjadi secara konsisten pada seluruh pengulangan eksperimen. Di sisi lain, peningkatan MAX_LEN dari 128 menjadi 256 menyebabkan kenaikan waktu pelatihan sebesar 37,7%, penggunaan memori GPU sebesar 16,3%, dan waktu inferensi sebesar 69,0%. Dengan demikian, hasil pengujian menunjukkan bahwa peningkatan maximum *sequence length* tidak selalu memberikan perbaikan kinerja yang sebanding dengan tambahan biaya komputasi.

Pengaruh Maximum *Sequence Length* terhadap *Truncation* dan *Word Coverage*

Dampak perubahan maximum sequence length dapat diidentifikasi melalui tingkat *truncation* dan cakupan kata yang dihasilkan. Pada konfigurasi MAX_LEN=64, terdapat 8 dari 1.464 *sequence* yang mengalami pemotongan, atau setara dengan 0,5464% dari total data latih. Kondisi ini berkaitan dengan karakteristik panjang *sequence* dalam *dataset* IDNer-News-2K serta proses tokenisasi *subword* menggunakan *tokenizer* IndoBERT. Meskipun rata-rata panjang kalimat dalam dataset relatif pendek, yaitu 20,81 token pada representasi kata, setiap kalimat memiliki panjang yang bervariasi. Selain itu, proses tokenisasi IndoBERT dapat memecah sebuah kata menjadi beberapa *subword* sehingga jumlah token yang dihasilkan dapat lebih besar daripada jumlah kata aslinya. Dengan mempertimbangkan penambahan token khusus [CLS] dan [SEP], konfigurasi MAX_LEN=64 hanya menyediakan 62 posisi untuk *subword* teks. Akibatnya, *sequence* yang menghasilkan lebih dari 62 subword akan mengalami *truncation*. Temuan ini menunjukkan bahwa pemilihan *maximum sequence length* perlu mempertimbangkan distribusi panjang input setelah tokenisasi, bukan hanya rata-rata jumlah kata pada *dataset*. Pengaruh *maximum sequence length* terhadap *truncation* dan *word coverage* dapat dilihat di Tabel 5.



Gambar 3. Dampak MAX_LEN terhadap Truncation

Tabel 5. *Truncation* dan *Coverage* pada Data Latih

Max Len	*Sequence*	*Sequence Terpotong*	*Truncation (%)*	*Word Coverage (%)*	*Token Coverage (%)*
64	1.464	8	0,5464	99,8215	99,7104
128	1.464	0	0	100	100
256	1.464	0	0	100	100
512	1.464	0	0	100	100

Berdasarkan Gambar 3, peningkatan nilai MAX_LEN dari 64 menjadi 128 secara keseluruhan mengeliminasi pemotongan *sequence*. Seluruh kata dan token dalam set pelatihan telah tercakup pada nilai

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

MAX_LEN sebesar 128, 256, dan 512. Penggunaan 128 token menunjukkan Panjang ini memadai untuk merepresentasikan seluruh sekuens dalam dataset tanpa menyebabkan kehilangan informasi akibat *truncation*.

Peningkatan *sequence length* dari 128 menjadi 256 atau 512 tidak memberikan tambahan informasi input baru pada model. Kondisi tersebut menunjukkan kecenderungan stagnasi performa model setelah penggunaan parameter MAX_LEN=128. Meskipun penggunaan parameter MAX_LEN=256 menghasilkan nilai *Entity* F1 yang sedikit lebih tinggi, peningkatan tersebut bukan disebabkan oleh perluasan cakupan data. Hal ini disebabkan tingkat *word coverage* pada kedua konfigurasi telah mencapai 100%.

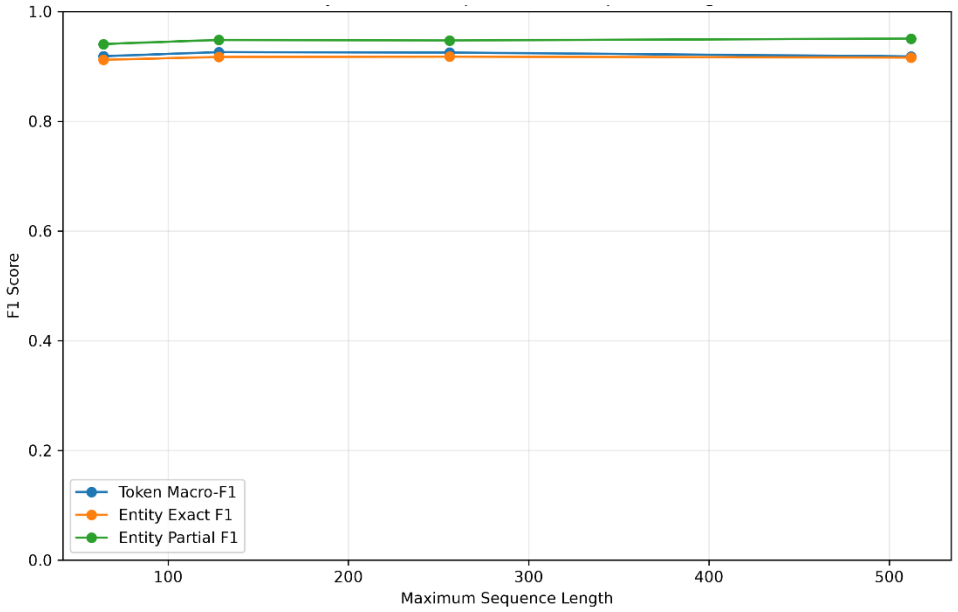
Pengaruh *Maximum Sequence Length* terhadap Kinerja NER

Gambar 4 menunjukkan bahwa hubungan antara *sequence length* dan performa NER. Peningkatan parameter MAX_LEN dari 64 menjadi 128 memberikan kenaikan pada skor *Entity* F1 sebesar kurang lebih 0,52 poin persentase dan skor Token *Macro*-F1 sebesar kurang lebih 0,76 poin persentase. Peningkatan tersebut terjadi bersamaan dengan penghilangan truncation pada parameter MAX_LEN=128.

Pada MAX_LEN=256, nilai *Entity* F1 mencapai angka tertinggi sebesar 0,9177, dengan peningkatan sebesar 0,0006 dibandingkan pada penggunaan MAX_LEN=128. Selain itu, nilai Token *Macro*-F1 pada penggunaan MAX_LEN=256 menunjukkan penurunan yang sedikit lebih rendah dibandingkan dengan MAX_LEN=128. Peningkatan *sequence length* menjadi 512 menyebabkan penurunan pada kedua metrik tersebut. Nilai entitas F1 mengalami penurunan sebesar 0,17 poin persentase dibandingkan dengan MAX_LEN=256, sementara nilai Token *Macro*-F1 menurun sebesar 0,71 poin persentase.

Temuan ini mengindikasikan terjadinya penurunan *diminishing return* setelah panjang input mampu mengakomodasi seluruh sekuens. Peningkatan panjang sekuens tidak selalu memberikan tambahan informasi kontekstual yang signifikan, terutama apabila seluruh teks dapat diproses sepenuhnya pada panjang 128. Sebaliknya, penggunaan padding yang lebih panjang menyebabkan model memproses tensor dengan dimensi yang lebih besar tanpa memberikan tambahan token yang bermakna pada sebagian besar sekuens.

Metrik *precision* dan *recall* menunjukkan pola yang serupa. Penggunaan parameter MAX_LEN=256 menghasilkan nilai *recall* tertinggi sebesar 0,9225, sementara parameter MAX_LEN=128 menunjukkan nilai precision yang sedikit lebih tinggi dibandingkan dengan parameter MAX_LEN=256. Konfigurasi 256 menunjukkan tingkat sensitivitas yang sedikit lebih tinggi dalam identifikasi entitas, namun peningkatan tersebut tidak signifikan jika dibandingkan dengan tambahan biaya komputasi yang diperlukan.



Gambar 4. Kinerja NER terhadap MAX_LEN

Pengaruh *Maximum Sequence Length* terhadap Efisiensi Komputasi

Tabel 6 menunjukkan perubahan efisiensi komputasi pada berbagai konfigurasi maximum sequence length. Peningkatan MAX_LEN menyebabkan bertambahnya waktu pelatihan dan inferensi, penggunaan memori GPU, serta penurunan *throughput*. Kondisi ini berkaitan dengan karakteristik arsitektur BERT yang menggunakan mekanisme self-attention untuk mempelajari hubungan antartoken dalam suatu *sequence*. Mekanisme *self-attention* membentuk matriks hubungan antartoken dengan ukuran yang meningkat secara kuadratik terhadap panjang sequence, yaitu $O(L^2)$, dengan L merupakan panjang input. Oleh karena itu, peningkatan MAX_LEN dapat memperbesar jumlah operasi komputasi dan kebutuhan memori selama proses pelatihan maupun inferensi.

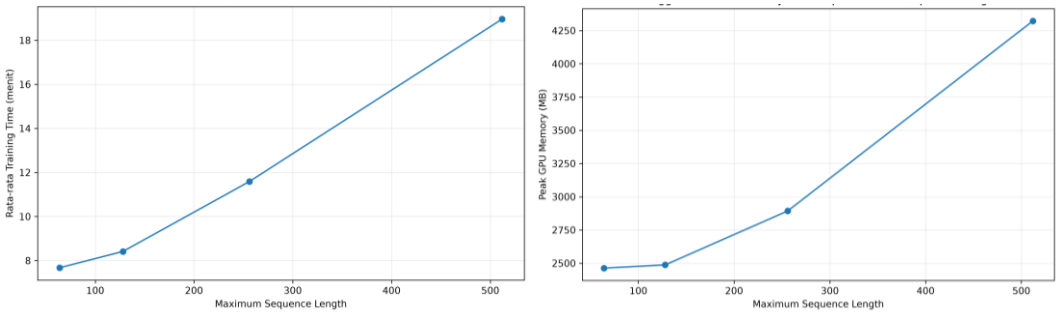
Selain mekanisme *self-attention*, penggunaan *fixed padding* pada eksperimen ini menyebabkan seluruh sequence disesuaikan dengan panjang maksimum yang ditentukan. Akibatnya, peningkatan MAX_LEN

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

memperbesar dimensi tensor yang diproses meskipun sebagian besar sequence memiliki panjang aktual yang lebih pendek. Keluaran IndoBERT selanjutnya diproses oleh BiLSTM untuk mempelajari dependensi sekuensial sebelum menghasilkan skor emisi bagi CRF. Dengan demikian, peningkatan panjang input tidak hanya memengaruhi proses komputasi pada IndoBERT, tetapi juga menambah pemrosesan pada lapisan sekuensial berikutnya.

Berdasarkan Tabel 6, MAX_LEN=64 menghasilkan beban komputasi paling rendah dengan rata-rata waktu pelatihan sebesar 7,68 menit dan throughput sebesar 1.294,21 token per detik. Peningkatan MAX_LEN menjadi 128 menyebabkan kenaikan waktu pelatihan sekitar 9,6% dan penggunaan memori GPU sekitar 1,0%. Pada konfigurasi tersebut, seluruh sequence pada data latih dapat diproses tanpa truncation sehingga cakupan input mencapai 100%. Sementara itu, peningkatan MAX_LEN menjadi 256 dan 512 menyebabkan tambahan biaya komputasi yang lebih besar, meskipun tidak lagi meningkatkan cakupan token pada data latih. Hasil ini menunjukkan bahwa pemilihan *maximum sequence length* perlu mempertimbangkan karakteristik panjang input agar peningkatan kebutuhan komputasi sebanding dengan informasi yang dipertahankan dan kinerja NER yang dihasilkan.

Konfigurasi MAX_LEN=64 merupakan pengaturan dengan beban komputasi paling ringan, yang menghasilkan rata-rata waktu pelatihan selama 7,68 menit dan tingkat throughput sebesar 1.294 token per detik. Peningkatan parameter MAX_LEN menjadi 128 menyebabkan kenaikan durasi pelatihan sebesar 9,6% dan penggunaan memori GPU sebesar 1%. Penyesuaian tersebut berhasil mengeliminasi proses *truncation* dan meningkatkan skor *Entity* F1.



Gambar 5. *Training Time* dan GPU *Memory* terhadap MAX_LEN

Gambar 5 menunjukkan grafik performa training time dan GPU *memory* terhadap MAX_LEN. Peningkatan nilai MAX_LEN dari 128 menjadi 256 menyebabkan kenaikan waktu pelatihan sebesar 37,7%, penggunaan memori GPU sebesar 16,3%, serta peningkatan waktu inferensi sebesar 69%. Pada periode yang sama, terjadi penurunan throughput sebesar 26,9%. Penambahan biaya tersebut hanya memberikan peningkatan pada skor *Entity* F1 sebesar kurang lebih 0,0006.

Dampak yang lebih signifikan ditunjukkan di MAX_LEN=512. Dibandingkan dengan parameter MAX_LEN=128, terdapat peningkatan waktu pelatihan sebesar 125,4%, peningkatan penggunaan memori GPU sebesar 73,8%, serta peningkatan waktu inferensi sebesar 230,6%. Sebaliknya, nilai throughput mengalami penurunan sebesar 55,2%. Namun, nilai Entity F1 tercatat sedikit lebih rendah dibandingkan dengan parameter MAX_LEN=128. Peningkatan sequence length mengakibatkan peningkatan beban komputasi yang melampaui perolehan peningkatan performa yang dihasilkan..

Tabel 6. Efisiensi Komputasi pada Berbagai Maximum Sequence Length

Max Len	Waktu *Training* (menit)	*Peak* GPU *Memory* (MB)	*Throughput* (token/s)	*Inference Time* (detik)
64	7,68	2.462,84	1.294,21	1,32
128	8,41	2.487,15	1.189,38	1,95
256	11,59	2.893,08	869,21	3,29
512	18,96	4.322,24	533,04	6,43

Diskusi

Secara keseluruhan, hasil penelitian menunjukkan bahwa pengaruh *maximum sequence length* terhadap IndoBERT-BiLSTM-CRF tidak hanya berkaitan dengan nilai F1, tetapi juga dengan cakupan input dan biaya komputasi. Pada MAX_LEN=64 masih terjadi *truncation* sekitar 0,55% dengan *word coverage* 99,82%, sehingga menghasilkan *Entity* F1 terendah sebesar 0,9119. Peningkatan menjadi MAX_LEN=128 menghilangkan seluruh truncation, menghasilkan *word coverage* dan token *coverage* sebesar 100%, serta meningkatkan *Entity* F1 menjadi 0,9171. Setelah titik ini, peningkatan panjang *sequence* tidak lagi menambah informasi input karena seluruh *sequence* telah terakomodasi.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

Untuk menjaga independensi evaluasi, penelitian ini mempertahankan pembagian *train, validation*, dan *test* sebagaimana tersedia pada *dataset* IDNer-News-2K. *Training s*et digunakan untuk pembelajaran parameter model, sedangkan *validation* set digunakan untuk menentukan *checkpoint* terbaik berdasarkan *Entity* F1. *Test set* digunakan untuk mengevaluasi kinerja akhir model. Analisis *truncation* dan *coverage* pada *test set* dilakukan untuk mendeskripsikan karakteristik input dan tidak digunakan sebagai dasar penentuan konfigurasi *maximum sequence* length. Dengan pemisahan fungsi tersebut, proses pemilihan model diarahkan agar tidak memanfaatkan informasi hasil evaluasi *test set.*

MAX_LEN=256 menghasilkan *Entity* F1 tertinggi sebesar 0,9177, tetapi hanya berbeda sekitar 0,0006 dibandingkan MAX_LEN=128. Sebaliknya, waktu training meningkat dari 8,41 menjadi 11,59 menit dan penggunaan GPU *memory* meningkat dari sekitar 2.487 MB menjadi 2.893 MB. Pada MAX_LEN=512, biaya komputasi meningkat lebih jauh dengan waktu *training* mencapai 18,96 menit dan penggunaan GPU *memory* sekitar 4.322 MB, sementara *Entity* F1 justru menurun menjadi 0,9159. Hasil ini menunjukkan adanya *diminishing return*, yaitu peningkatan panjang *sequence* setelah seluruh input tercakup tidak menghasilkan peningkatan performa yang sebanding dengan tambahan biaya komputasi.

Oleh karena itu, pemilihan *maximum sequence length* sebaiknya mempertimbangkan karakteristik panjang *sequence* pada *dataset*, bukan sekadar menggunakan batas maksimum *Transformer* seperti 512. Pada *dataset* IDNer-News-2K, MAX_LEN=128 memberikan keseimbangan terbaik antara kinerja dan efisiensi komputasi, sedangkan MAX_LEN=256 dapat dipilih apabila prioritas utama adalah memperoleh *Entity* F1 tertinggi tanpa mempertimbangkan tambahan biaya komputasi. Temuan ini menunjukkan bahwa panjang *sequence* yang lebih besar tidak selalu menghasilkan model NER yang lebih baik.

4. SIMPULAN

Penelitian ini menganalisis pengaruh *maximum sequence length* terhadap kinerja dan efisiensi komputasi model IndoBERT-BiLSTM-CRF pada NER Bahasa Indonesia menggunakan *dataset* IDNer-News-2K. Hasil menunjukkan bahwa peningkatan MAX_LEN dari 64 menjadi 128 mampu menghilangkan *truncation* dan mencapai 100% *word coverage* serta *token coverage*. MAX_LEN=256 menghasilkan *Entity* F1 tertinggi sebesar 0,9177, tetapi hanya sedikit lebih tinggi dibandingkan MAX_LEN=128 sebesar 0,9171. Sebaliknya, peningkatan panjang *sequence* menyebabkan kenaikan waktu pelatihan, penggunaan GPU *memory*, dan waktu inferensi serta penurunan *training throughput*. Pada MAX_LEN=512, biaya komputasi meningkat secara substansial sementara performa tidak mengalami peningkatan.

Berdasarkan hasil eksperimen, MAX_LEN=128 menunjukkan keseimbangan yang menguntungkan antara kinerja *Named Entity Recognition* dan efisiensi komputasi pada dataset IDNer-News-2K. Konfigurasi tersebut menghasilkan rata-rata Entity F1 sebesar 0,9171, mendekati MAX_LEN=256 yang mencapai 0,9177, tetapi dengan kebutuhan waktu pelatihan, memori GPU, dan waktu inferensi yang lebih rendah. Temuan ini menunjukkan bahwa peningkatan *maximum sequence length* tidak selalu diikuti peningkatan kinerja yang sebanding dengan tambahan biaya komputasi. Meskipun demikian, hasil penelitian ini terbatas pada karakteristik dataset IDNer-News-2K, arsitektur IndoBERT-BiLSTM-CRF, serta konfigurasi eksperimen yang digunakan. Oleh karena itu, MAX_LEN=128 tidak dapat secara langsung dianggap sebagai konfigurasi optimal untuk seluruh dataset dan domain NER Bahasa Indonesia tanpa pengujian lebih lanjut.

Penelitian ini memiliki beberapa keterbatasan. Pertama, evaluasi hanya dilakukan menggunakan dataset IDNer-News-2K yang berasal dari domain berita, sehingga hasilnya belum tentu merepresentasikan karakteristik teks dari domain lain. Kedua, penelitian hanya menggunakan satu arsitektur, yaitu IndoBERT-BiLSTM-CRF, dengan hiperparameter selain *maximum sequence length* dipertahankan tetap. Oleh karena itu, hasil eksperimen dapat dipengaruhi oleh konfigurasi hiperparameter yang digunakan. Ketiga, penggunaan tiga *random seed* masih membatasi analisis variabilitas hasil eksperimen. Penelitian selanjutnya dapat melibatkan *dataset* lintas domain, variasi arsitektur, dan pengulangan eksperimen yang lebih banyak untuk mengevaluasi konsistensi temuan.

Penelitian selanjutnya dapat mengevaluasi nilai *maximum sequence length* yang lebih rapat di sekitar titik optimal, seperti 96, 128, 160, dan 192, untuk memperoleh batas efisiensi yang lebih presisi. Eksperimen juga perlu dilakukan pada *dataset* NER Bahasa Indonesia dengan *sequence* yang lebih panjang dan domain yang berbeda untuk mengetahui apakah pola yang ditemukan pada IDNer-News-2K tetap konsisten. Selain itu, penelitian berikutnya dapat mengevaluasi strategi seperti *dynamic padding*, *length-based batching*, dan *sliding window* untuk mengurangi biaya komputasi tanpa kehilangan informasi, serta menggunakan pengujian statistik untuk menilai apakah perbedaan performa antar-konfigurasi benar-benar signifikan.

DAFTAR PUSTAKA

[1] E. Dave and A. Chowanda, “IPerFEX-2023: Indonesian Personal Financial Entity Extraction Using indoBERT-BiGRU-CRF model,” *Journal of Big Data*, vol. 11, no. 1, p. 139, Sep. 2024, doi: 10.1186/s40537-024-00987-6.

[2] Z. Zainuddin, - Mudassir, and Z. Tahir, “Entity Extraction in Indonesian Online News Using Named Entity Recognition (NER) with Hybrid Method Transformer, Word2Vec, Attention and Bi-LSTM,” *JOIV :*

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi

International Journal on Informatics Visualization, vol. 9, no. 3, p. 964, May 2025, doi: 10.62527/joiv.9.3.2902.

[3] S. O. Khairunnisa, Z. Chen, and M. Komachi, "Dataset Enhancement and Multilingual Transfer for Named Entity Recognition in the Indonesian Language," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 6, pp. 1–21, Jun. 2023, doi: 10.1145/3592854.

[4] N. Istiqomah and F. Novika, "Comparative Performance of IndoBERT and IndoLEM Baseline Models for Post-Disaster Health Information Extraction from Indonesian Online News," *Journal of Computer Science and Informatics Engineering*, vol. 4, no. 3, pp. 158–174, Jul. 2025, doi: 10.55537/cosie.v4i3.1174.

[5] A. Harnoune, M. Rhanoui, M. Mikram, S. Yousfi, Z. Elkaimbillah, and B. El Asri, "BERT Based Clinical Knowledge Extraction for Biomedical Knowledge Graph Construction and Analysis," *Computer Methods and Programs in Biomedicine Update*, vol. 1, p. 100042, 2021, doi: 10.1016/j.cmpbup.2021.100042.

[6] W. Sun, S. Liu, Y. Liu, L. Kong, and Z. Jian, "Named Entity Recognition Networks Based on Syntactically Constrained Attention," *Applied Sciences*, vol. 13, no. 6, p. 3993, Mar. 2023, doi: 10.3390/app13063993.

[7] R. Catelli, F. Gargiulo, V. Casola, G. De Pietro, H. Fujita, and M. Esposito, "Crosslingual named entity recognition for clinical de-identification applied to a COVID-19 Italian data set," *Applied Soft Computing*, vol. 97, p. 106779, Dec. 2020, doi: 10.1016/j.asoc.2020.106779.

[8] A. Roy, M. Saffar, A. Vaswani, and D. Grangier, "Efficient Content-Based Sparse Attention with Routing Transformers," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 53–68, Feb. 2021, doi: 10.1162/tacl_a_00353.

[9] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient Transformers: A Survey," *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–28, Jun. 2023, doi: 10.1145/3530811.

[10] A. van Engelenhoven, N. Strisciuglio, and E. Talavera, "CAST: Clustering self-Attention using Surrogate Tokens for Efficient Transformers," *Pattern Recognition Letters*, vol. 186, pp. 30–36, Oct. 2024, doi: 10.1016/j.patrec.2024.08.024.

[11] M. Yang, B. Namoano, M. Farsi, and J. Ahmet Erkoyuncu, "Named Entity Recognition in Aviation Products Domain Based on BERT," *IEEE Access*, vol. 12, pp. 189710–189721, 2024, doi: 10.1109/ACCESS.2024.3516390.

[12] S. O. Khairunnisa, A. Imankulova, and M. Komachi, "Towards a Standardized Dataset on Indonesian Named Entity Recognition," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 64–71. doi: 10.18653/v1/2020.aacl-srw.10.

[13] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.00677.

[14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[15] X. Ma and E. Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 1064–1074. doi: 10.18653/v1/P16-1101.

Copyright (c) HOAQ (High Education of Organization Archive Quality) : Jurnal Teknologi Informasi