MODEL *NAMED ENTITY RECOGNITION* (NER) DALAM BAHASA INDONESIA DENGAN MENGGUNAKAN BiLSTM-CNNs

LAPORAN TUGAS AKHIR

Oleh:

Sahrul

105216044

FAKULTAS SAINS DAN ILMU KOMPUTER

PROGRAM STUDI ILMU KOMPUTER

UNIVERSITAS PERTAMINA

OKTOBER 2020

Hak Cipta milik Universitas Pertamina *Dilindungi Undang-Undang* *© Copyright of Universitas Pertamina*

1. Dilarang mengutip karya tulis ini, kecuali:
   a. menyebutkan sumber sesuai kaidah kecendekiaan;
   b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
   c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

MODEL *NAMED ENTITY RECOGNITION* (NER) DALAM BAHASA INDONESIA DENGAN MENGGUNAKAN BiLSTM-CNNs

LAPORAN TUGAS AKHIR

Oleh:

Sahrul

105216044

FAKULTAS SAINS DAN ILMU KOMPUTER

PROGRAM STUDI ILMU KOMPUTER

UNIVERSITAS PERTAMINA

OKTOBER 2020

Hak Cipta milik Universitas Pertamina *Dilindungi Undang-Undang* *© Copyright of Universitas Pertamina*

1. Dilarang mengutip karya tulis ini, kecuali:
   a. menyebutkan sumber sesuai kaidah kecendekiaan;
   b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
   c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

Universitas
Pertamina

LEMBAR PENGESAHAN

Judul Tugas Akhir	:	Model *Named-Entity Recognition* (NER) dalam Bahasa Indonesia dengan Menggunakan BiLSTM-CNNs
Nama Mahasiswa	:	Sahrul
Nomor Induk Mahasiswa	:	105216044
Program Studi	:	Ilmu Komputer
Fakultas	:	Sains dan Ilmu Komputer
Tanggal Lulus Sidang Tugas Akhir	:	Senin, 7 September 2020

Jakarta, 17 September 2020

MENGESAHKAN

Pembimbing 1:

Meredita Susanty, M.Sc.

NIP. 116020

MENGETAHUI,

Ketua Program Studi

Muhamad Koymatu, Ph.D.

NIP. 116108

Universitas Pertamina

Hak Cipta milik Universitas Pertamina

Dilindungi Undang-Undang

© Copyright of Universitas Pertamina

1. Dilarang mengutip karya tulis ini, kecuali:
   a. menyebutkan sumber sesuai kaidah kecendekiaan;
   b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
   c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

LEMBAR PERNYATAAN

Dengan ini saya menyatakan bahwa Tugas Akhir berjudul Model *Named-Entity Recognition (NER)* dalam Bahasa Indonesia dengan Menggunakan BiLSTM-CNNs ini adalah benar-benar merupakan hasil karya saya sendiri dan tidak mengandung materi yang ditulis oleh orang lain kecuali telah dikutip sebagai referensi yang sumbernya terlah dituliskan secara jelas dengan kaidah penulisan karya ilmiah.

Apabila dikemudian hari adanya kecurangan dalam karya ini, saya bersedia menerima sanksi dari Universitas Pertamina sesuai dengan peraturan yang berlaku.

Demi pengembangan ilmu pengetahuan, saya menyetujui untuk memberikan kepada Universitas Pertamina hak bebas royalti noneksklusif *(non-exclusive royalty-free right)* atas Tugas Akhir ini beserta perangkat yang ada. Dengan hak bebas royalti noneksklusif ini Universitas Pertamina berhak menyimpan, mengalih media/format-kan, mengelola dalam bentuk pangkatan data (*database*), merawat, dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 17 September 2020

Yang membuat pernyataan,

Sahrul

Hak Cipta milik Universitas Pertamina
Dilindungi Undang-Undang
© Copyright of Universitas Pertamina
Pertamina
1. Dilarang mengutip karya tulis ini, kecuali:
a. untuk kepentingan sumber sesuai kaidah kecendekiaan;
b. hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

ABSTRAK

Sahrul. 105216044. Model Named-Entity Recognition (NER) dalam Bahasa Indonesia dengan Menggunakan BiLSTM-CNNs.

Named-Entity Recognition (NER) merupakan bagian dari riset *Natural Language Processing* (NLP) yang digunakan untuk mengekstrak informasi seperti nama orang, organisasi, lokasi, dan waktu. NER telah digunakan di banyak bidang pekerjaan salah satunya pada pengembangan *chatbot*. Penggunaan NLP dan *machine learning* (ML) pada *chatbot* dapat membuat *chatbot* lebih cerdas dengan analisis personal yang lebih baik ke pengguna. Tujuan dari penelitian ini adalah untuk membuat model NER dalam bahasa Indonesia dengan menggunakan arsitektur *model Bidirectional Long-Short-Term-Memory* (BiLSTM) dan *Convolutional Neural Networks* (CNNs). Berbeda dengan penelitian sebelumnya, model ini hanya menggunakan *word-level embedding* pada lapisan CNNs untuk menjaga agar model tetap sederhana. *Named-entities* (NEs) yang digunakan pada penelitian ini hanya terbatas pada nama orang, organisasi, lokasi, kuantitas, dan waktu dengan menggunakan format pelabelan BILOU. Performa model yang dibangun diukur dengan menggunakan metrik evaluasi *f1 score* dengan rataan mikro. Model BiLSTM-CNNs + *pretrained* word2vec *embedding* menghasilkan performa yang sangat baik dibanding model lainnya dengan nilai *f1 score* rataan mikro 73.37%.

Kata kunci: *Natural Language Processing* (NLP), *Named-Entity Recognition* (NER), *Bidirectional Long Short-Term Memory* (BiLSTM), *Convolutional Neural Networks* (CNNs)

Hak Cipta milik Universitas Pertamina
Dilindungi Undang-Undang
© Copyright of Universitas Pertamina
1. Dilarang mengubah, mendistribusikan, dan menjual karya tulis ini, kecuali:
a. menyebutkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

ABSTRACT

Sahrul. 105216044. Named-Entity Recognition (NER) Model in Indonesian Language using BiLSTM-CNNs.

Named-Entity Recognition (NER) is part of Natural Language Processing (NLP) research used to extract information such as names of the people, organizations, and places. NER has been used in many fields of work, one of which is in chatbot development. NLP and machine learning approaches enable a smarter chatbot with better personal analysis to users. This research builds a NER model in Indonesian Language using Bidirectional Long-Short-Term-Memory (BiLSTM) and Convolutional Neural Networks (CNNs) model architecture. Unlike the former research, this model only uses word-level embedding in the CNNs layer to keep the model simple. The Named-Entities (NEs) used in this study are limited to the name of the person, organization, location, quantity, and time using the BILOU labeling format. The performance of the model built is measured using the f1 score evaluation metric with the micro average. The BiLSTM-CNNs + *pretrained* word2vec embedding model provides good performance compared to other models with an f1 score of 71.37%.

Keywords: Natural Language Processing (NLP), Named-Entity Recognition (NER), Bidirectional Long Short-Term Memory (BiLSTM), Convolutional Neural Networks (CNNs)

Hak Cipta milik Universitas Pertamina
Ditindungi Undang-Undang
© Copyright of Universitas Pertamina
1. Dilarang mengutip karya tulis ini, kecuali:
a. Mencantumkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan penulis nikmat-Nya sehingga penulis dapat menyelesaikan laporan tugas akhir ini dengan tepat waktu. Tanpa ridho dan nikmat-Nya tentu penulis tidak akan sanggup untuk menyelesaikan proses penelitian sampai penulisan laporan tugas akhir ini.

Pertama-tama, penulis hendak mengucapkan terima kasih atas bimbingan dan dukungan yang diberikan oleh Ibu Meredita Susanty selaku pembimbing dalam mengerjakan tugas akhir ini. Selanjutnya, penulis ingin mengucapkan terima kasih kepada dosen Universitas Pertamina khususnya dosen program Studi Ilmu Komputer atas bimbingan dan ilmu yang telah diberikan selama kuliah di Universitas Pertamina. Tak lupa pula penulis ucapkan terima kasih kepada rekan-rekan Ilmu Komputer 2016 dan seluruh pihak yang telah berperan dalam penelitian tugas akhir ini yang tidak bisa penulis sebutkan satu persatu.

Akhir kata, penulis memohon maaf apabila penulisan laporan tugas akhir ini masih jauh dari kata sempurna, oleh karena itu penulis memohon saran untuk membantu penulis agar dapat menulis lebih baik lagi.

Jakarta, 8 Agustus 2020

Sahrul

Hak Cipta milik Universitas Pertamina
Ditindungi Undang-Undang
© Copyright of Universitas Pertamina
Universitas Pertamina
1. Dilarang mengutip karya tulis ini, kecuali:
a. menebutkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

DAFTAR ISI

ABSTRAK i

ABSTRACT ii

KATA PENGANTAR iii

DAFTAR ISI iv

DAFTAR TABEL vi

DAFTAR GAMBAR vii

BAB I PENDAHULUAN 1

- **1.1. Latar Belakang** 1
- **1.2. Rumusan Masalah** 2
- **1.3. Batasan Masalah** 2
- **1.4. Tujuan Penelitian** 2
- **1.5. Manfaat Penelitian** 2

BAB II TINJAUAN PUSTAKA 3

- **2.1. Named-Entity Recognition** 3
- **2.2. Machine Learning** 4
- **2.2. Neural Networks** 5
  - **2.2.1. Reccurent Neural Networks** 6
  - **2.2.2. Long Short-Term Memory** 6
  - **2.2.3. Convolutional Neural Networks** 8
- **2.3. Word Embedding** 9
- **2.4. One-Hot-Encoding** 10
- **2.5. Learning Curve** 10
- **2.6. Metrik Evaluasi** 11
- **2.7. Penelitian NER Terdahulu** 12

BAB III METODE PENELITIAN 14

- **3.1. Persiapan Data** 14
- **3.2. Pra-pemrosesan Data** 14
- **3.3. Pemodelan** 15
- **3.4. Evaluasi** 16

BAB IV HASIL DAN PEMBAHASAN 17

- **4.1. Persiapan Data** 17
- **4.2. Pra-pemrosesan Data** 18
- **4.3. Pemodelan** 19
  - **4.3.1. Embedding** 19
  - **4.3.2. BiLSTM** 21
  - **4.3.3. BiLSTM-CNNs** 23
  - **4.3.4. BiLSTM-CNNs + multi-sequence n-gram** 25
  - **4.3.5. Optimisasi Hiper-parameter** 28

Universitas Pertamina

Hak Cipta milik Universitas Pertamina

Perpustakaan Universitas Pertamina

1. Dilarang mengutip karya tulis ini sebagian atau seluruh:
   a. mencantumkan dan menyebutkan sumber sesuai kaidah kecendekiaan;
   b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
   c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

4.4. Analisis **30**
BAB V KESIMPULAN DAN SARAN **32**
5.1. Kesimpulan **32**
5.2. Saran **32**
DAFTAR PUSTAKA **33**
LAMPIRAN **36**

Hak Cipta milik Universitas Pertamina Dilindungi Undang-Undang © Copyright of Universitas Pertamina

1. Dilarang mengutip karya tulis ini, kecuali:
   a. menyebutkan sumber sesuai kaidah kecendekiaan;
   b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
   c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

DAFTAR TABEL

Tabel 4.1. Jumlah kemunculan label 17
Tabel 4.2. Nilai hiper-parameter awal pada model 19
Tabel 4.3. Nilai f1 score setiap pemilihan panjang sequence pada model BiLSTM..... 22
Tabel 4.4. Nilai f1 score setiap pemilihan panjang sequence pada model BiLSTM-CNNs 24
Tabel 4.5. Nilai f1 score setiap pemilihan panjang sequence pada model BiLSTM-CNNs 26
Tabel 4.6. Optimasi nilai hiper-parameter pada model BiLSTM + word2vec..... 28
Tabel 4.7. Optimasi nilai hiper-parameter pada model BiLSTM-CNNs + word2vec 28
Tabel 4.8. Optimasi nilai hiper-parameter pada model BiLSTM-CNNs + multi-sequence..... 29
Tabel 4.9. Perbandingan performa model pada data uji dengan nilai hiper-parameter terbaik..... 30
Tabel 4.10. Perbandingan performa model pada rata-rata nilai f1 score setiap NEs 30
Tabel 4.11. Perbandingan performa model pada rata-rata nilai f1 score setiap posisi token 30

Hak Cipta milik Universitas Pertamina
Dilindungi Undang-Undang
© Copyright of Universitas Pertamina
Universitas Pertamina
1. Dilarang mengutip karya tulis ini, kecuali:
a. menyebutkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

DAFTAR GAMBAR

Gambar 2.1. Model neuron pada NNs [17] 5
Gambar 2.2. Perceptron [17] 6
Gambar 2.3. LSTM unit [25] 6
Gambar 2.4. Bi-directional RNNs [28] 8
Gambar 2.5. Lapisan CNNs untuk klasifikasi kalimat [29] 9
Gambar 2.6. Lapisan CNNs untuk character-level embedding [30] 9
Gambar 2.7. Learning curve pada proses pelatihan artificial neural network [38] 10
Gambar 2.8. Confusion-matrix untuk klasifikasi biner [35] 11
Gambar 3.1. Alur kerja penelitian 14
Gambar 3.2. Arsitektur model BiLSTM-CNNs yang digunakan 15
Gambar 4.1. Distribusi panjang token setiap observasi data 18
Gambar 4.2. Kata-kata yang mirip dengan “Makassar”: (biru) kata yang sangat mirip dan 20
Gambar 4.3. Arsitektur model BiLSTM 21
Gambar 4.4. Perubahan nilai loss: trainable (kiri) dan word2vec (kanan) 21
Gambar 4.5. Perubahan nilai f1 score: trainable (kiri) dan word2vec (kanan) 21
Gambar 4.6. Confusion matrix pada data uji pada model BiLSTM 22
Gambar 4.7. Arsitektur model BiLSTM-CNNs 23
Gambar 4.8. Perubahan nilai loss: trainable (kiri) dan word2vec (kanan) 23
Gambar 4.9. Perubahan nilai f1 score: trainable (kiri) dan word2vec (kanan) 23
Gambar 4.10. Confusion matrix pada data uji pada model BiLSTM-CNNs 24
Gambar 4.11. Arsitektur model BiLSTM-CNNs + multi-sequence n-gram 25
Gambar 4.12. Perubahan nilai loss: trainable (kiri) dan word2vec (kanan) 25
Gambar 4.13. Perubahan nilai f1 score: trainable (kiri) dan word2vec (kanan) 26
Gambar 4.14. Confusion matrix pada data uji pada model BiLSTM-CNNs + multi-sequence n 27

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
Dilindungi Undang-Undang © Copyright Universitas Pertamina
1. Dilarang mengutip kata tulis ini, kecuali:
a. mencebukan sumber sesai kidan keendekian;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

Universitas
Pertamina

BAB I
PENDAHULUAN

Bab ini menjelaskan tentang pendahuluan dan latar belakang penelitian yang telah dilakukan. Sub-bab 1.1 berisi alasan mengapa topik yang diangkat pada penelitian ini penting untuk dilakukan. Pada sub-bab 1.2 dan 1.3 diuraikan rumusan dan batasan masalah yang dibahas terkait dengan sub-bab sebelumnya. Sub-bab 1.4 dan 1.5 berisi formulasi tentang tujuan penelitian yang telah dilakukan dan kontribusi penelitian ini terhadap perkembangan ilmu pengetahuan dan teknologi (IPTEK).

1.1. Latar Belakang

Named-Entity Recognition (NER) merupakan sub-bagian dari riset *Natural Language Processing* (NLP) yang termasuk dalam bidang *Artificial Intelligence* (AI). NER digunakan untuk mengestrak informasi dari teks dengan cara mengidentifikasi entitas seperti nama orang, organisasi, lokasi, waktu, dan entitas lainnya [1]. NER telah diaplikasikan di banyak bidang pekerjaan salah satunya pada pengembangan program *automated question answering* seperti *chatbot* [2]. *Chatbot* merupakan sebuah program komputer yang dapat merespon pertanyaan dan berkomunikasi dengan pengguna menggunakan bahasa natural layaknya manusia sungguhan [3]. Penelitian [2] menggunakan NER untuk membantu *chatbot* mengenali pola entitas bahasa Indonesia yang beraneka ragam. Penggunaan NLP dan *Machine Learning* (ML) dapat membuat *chatbot* lebih cerdas dengan analisis personal yang lebih baik ke pengguna [4].

Saat ini, penelitian terkait NER telah banyak dilakukan dalam berbagai tujuan dan metode yang digunakan [5, 6, 7]. Metode yang digunakan pun beragam, mulai dari *rule-based* sampai dengan penggunaan algoritma ML yang kompleks. Metode *rule-based* menggunakan aturan yang telah didefinisikan berdasarkan pengetahuan linguistik dengan analisis yang dilakukan pada tingkatan sintaksis dan semantik [8]. Metode tersebut memiliki keterbatasan karena kita harus mendefinisikan aturan sebanyak mungkin untuk memperoleh hasil yang optimal [9]. Untuk mengatasi keterbatasan tersebut, kita dapat menggunakan pendekatan ML untuk mempelajari pola dari data dengan hanya menyediakan *data set* yang memadai. Algoritma ML yang dapat digunakan salah satunya adalah algoritma *Naïve Bayes*. Algoritma tersebut melakukan klasifikasi dengan cara menghitung probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari *data set* yang diberikan [10]. Pendekatan yang juga banyak digunakan belakangan ini adalah dengan menggunakan *Deep Learning* (DL) untuk mengenali pola entitas dalam kalimat [6, 30]. Algoritma DL merupakan bagian dari algoritma ML yang *robust* digunakan untuk memodelkan permasalahan yang kompleks seperti NER [11].

Model yang dikembangkan pada penelitian ini terinspirasi dari [6, 30], tetapi dengan model yang lebih sederhana. Model pembelajaran yang digunakan mirip dengan [6], yaitu dengan menggunakan model *Bidirectional Long Short-Term Memory* (BiLSTM) dan *Convolutional Neural Networks* (CNNs). Lapisan CNNs yang digunakan pada [6] menggunakan lima lapisan konvolusi satu dimensi dengan ukuran kernel 2, 4, 6, 8, dan 10. Penggunaan lapisan konvolusi yang terlalu banyak dapat menyebabkan model *overfit* terhadap data latih dan ukuran kernel yang terlalu besar dapat menghilangkan informasi *sequence* dari data yang digunakan sebagai masukan pada lapisan LSTM. Model yang diusulkan pada penelitian ini hanya menggunakan satu lapisan konvolusi dan dengan ukuran kernel lebih rendah yang diperoleh dari proses optimasi hiper-parameter. Performa model NER yang dibangun dievaluasi dengan mengukur kemampuan model untuk melakukan prediksi.

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
1. Dilarang mengutip karya tulis sebagian atau seluruh karya tulis ini dalam bentuk apapun, termasuk dengan cara mengutip, menyebutkan sumber sesuatu hal yang wajar dan tidak merugikan kepentingan Universitas Pertamina
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
© Copyright © Universitas Pertamina

Kemampuan prediksi model diukur dengan menggunakan metrik *f1 score* pada setiap label dan dengan *f1 score* rataan mikro pada keseluruhan label.

1.2. Rumusan Masalah

Rumusan masalah untuk penelitian ini adalah sebagai berikut:

1. Bagaimana memodelkan model BiLSTM-CNNs agar dapat mempelajari *data set* dengan baik?
2. Bagaimana pengaruh penggunaan lapisan konvolusi terhadap performa model BiLSTM-CNNs?

1.3. Batasan Masalah

Batasan masalah yang diangkat pada penelitian ini adalah sebagai berikut:

1. Pengembangan model NER hanya terbatas pada sumber data publik yang digunakan.
2. Model NER yang dikembangkan hanya dievaluasi selama proses pengembangan dengan metrik *f1 score* rataan mikro.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Terciptanya metode pengembangan model NER dalam bahasa Indonesia.
2. Terciptanya model NER yang dapat digunakan dalam aplikasi yang memerlukan pemahaman bahasa natural oleh mesin dalam bahasa Indonesia.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini terhadap perkembangan Ilmu Pengetahuan dan Teknologi (IPTEK) adalah sebagai berikut:

1. Sebagai bahan rujukan terkait dengan pengembangan model NER dalam bahasa Indonesia.
2. Model yang dihasilkan dapat digunakan oleh aplikasi yang memerlukan pemahaman bahasa natural oleh mesin dalam bahasa Indonesia.

Hak Cipta milik Universitas Pertamina (Universitas Pertamina)
Dilarang mengutip karya tulis ini, kecuali:
a. menyebutkan sumber sesuai kaidah keilmiahan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
© Copyright of Universitas Pertamina

Universitas
Pertamina

BAB II
TINJAUAN PUSTAKA

Bab ini menguraikan tinjauan pustaka yang terkait dengan penelitian ini. Sub-bab 2.1 menjelaskan tentang NER. Pada sub-bab 2.2 akan dijelaskan tentang ML dan jenis model pembelajaran yang digunakan dalam pengembangan model NER. Sub-bab 2.3 dan 2.4 dijelaskan tentang teknik pra-pemrosesan yang digunakan pada data teks. Sub-bab 2.5 diuraikan tentang metrik evaluasi pada permasalahan klasifikasi. Sub-bab 2.6 dijelaskan beberapa penelitian terdahulu dan perbedaan penelitian tersebut dengan yang dilakukan pada penelitian ini.

2.1. Named-Entity Recognition

Named-Entities (NEs) merupakan sebuah kata atau frasa yang dengan jelas dapat mengidentifikasi satu item entitas ke item entitas lainnya yang memiliki atribut serupa [12]. Contoh NEs yang sering ditemukan adalah nama organisasi, orang, lokasi, waktu, dan lain sebagainya. Istilah NEs sekarang banyak digunakan pada bidang *natural language processing* (NLP) yang pada awalnya muncul pada *The Sixth Message Undestanding Conference* (MUC-6) [13]. Pada saat itu, konferensi MUC ke 6 fokus pada *information retrieval* terkait dengan aktivitas perusahaan dan pertahanan dari artikel-artikel surat kabar. Dalam proses tersebut, orang-orang menyadari bahwa penting untuk mengenali informasi entitas seperti nama orang, organisasi, lokasi, waktu, dan lain sebagainya. Proses identifikasi entitas tersebut kemudian diakui sebagai salah satu bagian dari *information retrieval* dan disebut dengan *Named-Entity Recognition* (NER).

NER telah banyak digunakan pada data teks dengan menggunakan metode *supervised learning* untuk mengklasifikasikan NEs sesuai domain penelitian yang dilakukan. Pelatihan model *supervised learning* memerlukan *data set* yang dilengkapi dengan label. Terdapat beberapa pendekatan yang dapat digunakan untuk melabeli data NER, dua diantaranya adalah skema pendekatan yang digunakan pada MUC-6 dan CoNLL 2009 [14].

Secara umum, teknik pelabelan pada NER disebut dengan *sequence labeling* dengan memberi label setiap NEs dalam kalimat atau frasa tertentu [39]. Pada MUC-6, NEs dibagi menjadi 3 jenis, yaitu ENAMEX yang terdiri dari NEs *person*, *organization*, *location*; TIMEX yang terdiri dari NEs *date* dan *time*; NUMEX yang terdiri dari NEs *money*, *percentage*, dan *quanity*. Contoh pelabelan *data set* (dalam bahasa Inggris) dengan menggunakan label entitas tersebut adalah sebagai berikut [13]:

The `<ENAMEX TYPE="LOCATION">` U.K `</ENAMEX>` satellite television broadcaster said its subscribers base grew `<NUMEX TYPE="PERCENT">` 17.5 percent `</ENAMEX>` during `<TIMEX TYPE="DATE">` the past year `</ENAMEX>` to 5.35 million.

Pada CoNLL, penelitian [14] fokus untuk membandingkan dua skema pelabelan yang paling populer yaitu BIO dan BILOU. Format BIO membagi NEs menjadi ***B****eginning* untuk token yang terletak di awal, ***I****nside* untuk token setelah token pertama apabila NEs memiliki lebih dari satu token, dan ***O****utside* untuk token yang tidak termasuk NEs yang telah didefinisikan. Format BILOU memberi informasi mengenai letak token lebih detail dengan hanya menambahkan ***L****ast* untuk token terakhir pada NEs multi token dan ***U****nit* untuk token tunggal.

Namun, pelabelan data NER tidak harus mengikuti format pelabelan MUC-6 dan CoNLL, tergantung pada domain penelitian yang dilakukan. Sebagai contoh, NER yang dilakukan pada penelitian dengan data dari dokumen biologi [15], NEs dibagi menjadi CELL COMPONENT yang terdiri dari

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
c. pengutipan tidak merugikan Universitas Pertamina.
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
a. mencantumkan sumber dan kepentingan;
Dilarang mengutip karya tulis in kecuali:
Hak Cipta milik Universitas Pertamina

Membran, *Nukleus*, *Ribosom*; PROTEIN yang terdiri dari *Lipoprotein* dan *Enzim*; RNA yang terdiri dari *Rrna* dan NEs lain yang didefinisikan di penelitian tersebut.

2.2. Machine Learning

Machine Learning (ML) merupakan salah satu cabang ilmu statistik terapan dengan lebih menekankan penggunaan algoritma pada komputer dengan kapasitas komputasi yang besar [16]. Dengan kemampuan untuk mempelajari *data set*, model ML dapat melakukan tugas tertentu yang sulit untuk dilakukan dengan program komputer seperti biasa. Sebagai contoh, model ML dapat dengan mudah membedakan foto anjing dan kucing dalam sebuah gambar apabila dilatih dengan *data set* yang cukup banyak dibandingkan dengan program komputer biasa. Program yang berbasis ML melakukan prediksi berdasarkan variabel-variabel prediktor untuk mengekstrak pola dari *data set* sehingga sangat bergantung pada kualitas dari *data set* yang digunakan [17].

Terdapat beberapa jenis pekerjaan yang dapat diselesaikan dengan menggunakan model ML. Beberapa jenis pekerjaan tersebut adalah sebagai berikut [16].

1. **Klasifikasi:** Pada jenis ini, model ML menentukan kelas atau kategori dari sebuah objek data berdasarkan masukan pada model.
2. **Regresi:** Jenis ini mirip dengan klasifikasi, tetapi luaran yang dihasilkan bukan merupakan kelas atau kategori, tetapi angka yang kontinu.
3. **Pengelompokan:** Pada jenis ini, model ML menentukan kelompok dari sebuah objek data berdasarkan dengan kemiripan karakteristik antara objek data yang lain.

Data yang digunakan untuk membuat program berbasis ML biasanya dibagi menjadi data latih, data validasi, dan data uji. Selama proses pelatihan dengan data latih, *hiper-parameter* dari model disesuaikan agar model dapat melakukan prediksi dengan baik. Selama proses pelatihan model ML, data validasi digunakan untuk memastikan bahwa performa model yang sedang dilatih tidak bias terhadap data latih dengan membandingkan hasil prediksi model pada data latih dengan data validasi. Setelah proses pelatihan, model yang telah dilatih diuji kembali dengan melakukan prediksi pada data uji yang tidak pernah dilihat sebelumnya.

Secara umum, berdasarkan ketersediaan informasi dari data, ML dapat dibedakan menjadi tiga kategori di bawah ini:

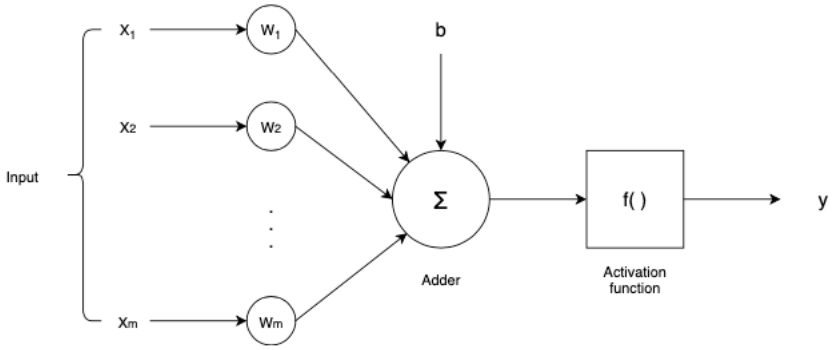
1. ***Supervised Learning***: Pada kategori ini, setiap observasi pada *data set* mempunyai variabel-variabel prediktor sebagai masukan dan label atau target sebagai luaran. Jenis pekerjaan klasifikasi dan regresi dapat dikategorikan sebagai *supervised learning*.
2. ***Unsupervised Learning*:** Tidak seperti *supervised learning*, pada kategori ini *data set* tidak memiliki informasi label atau target. Jenis pekerjaan pengelompokan dapat dikategorikan sebagai *unsupervised learning*.
3. ***Reinforcement Learning*:** Pada kategori ini, algoritma model ML tidak hanya belajar dari *data set*, tetapi juga berinteraksi dengan lingkungan dengan tujuan tertentu.

Pada sub-bab selanjutnya akan dibahas mengenai jenis-jenis model ML yang terkait dengan penelitian yang dilakukan.

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
1. Dilarang mengutip karya tulis ini, sebagian atau seluruh
a. menyebutkan sumber sesuai dengan kaidah ilmiah
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

2.2. Neural Networks

Neural Networks (NNs) merupakan kumpulan metode dalam ML yang dirancang untuk memodelkan bagaimana otak melakukan tugas tertentu [18, 19]. Otak adalah sistem yang sangat kompleks yang memiliki kemampuan untuk melakukan perhitungan tertentu untuk memproses informasi. Otak dapat membangun aturannya sendiri untuk melakukan tugas-tugas tertentu melalui proses pembelajaran dari waktu ke waktu. Oleh karena itu, otak menarik untuk digunakan sebagai acuan dalam merancang model untuk mengolah data yang kompleks.



Gambar 2.1. Model neuron pada NNs [17]

NNs menggunakan sel komputasi yang masif dan saling terkoneksi satu sama lain yang disebut *neuron*. Ada tiga elemen utama pada *neuron* seperti yang ditunjukkan pada Gambar 2.1 di atas.

1. Kumpulan ***Input*** yang merupakan masukan NNs dengan bobot masing-masing.
2. Sebuah ***Adder*** yang digunakan untuk menjumlahkan semua *inputs* sesuai dengan bobot masing-masing.
3. Sebuah ***Activation Function*** yang digunakan untuk membatasi luaran dari NNs dalam rentang tertentu.

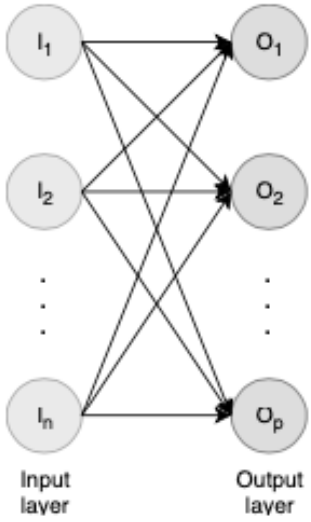
Secara matematis, proses komputasi pada *neuron* ditulis sebagai berikut [19].

$$y = f(b + \sum_{j=i}^{m} w_j x_j) \tag{2.1}$$

dimana $x_1, x_2, \dots, x_m$ adalah masukan; $w_1, w_2, \dots, w_m$ adalah bobot; b adalah bias; $f(\dots)$ adalah fungsi aktivasi; dan y adalah luaran dari *neuron*.

Arsitektur NNs yang sederhana dikenal dengan *feed-forward NNs*. Perceptron merupakan jenis *feed-forward NNs* paling sederhana yang hanya terdiri dari satu lapisan dari kumpulan p *neurons* dan terhubung dengan kumpulan n *inputs* seperti yang diilustrasikan pada Gambar 2.2.

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
Dilindungi Undang-Undang
© Copyright Universitas Pertamina
1. Dilarang mengutip karya tulis ini, kecuali:
a. menyebutkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.



Gambar 2.2. Perceptron [17]

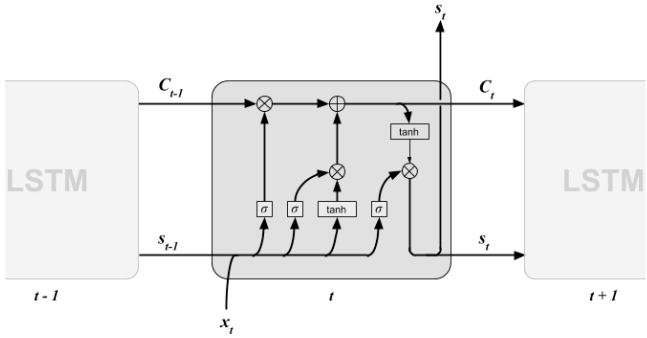
2.2.1. Reccurent Neural Networks

Reccurent neural networks (RNNs) terdiri dari *backward* atau *feedback connection*, disamping *feedforward connection* pada umumnya [16, 18]. Tipe NNs tersebut cocok digunakan untuk mengolah data sekuensial, seperti data *time-series*, teks, dan DNA. Setiap data tidak dapat saling terkait dengan data lainnya. Sebagai contoh, pada data *time-series*, luaran pada waktu t tidak hanya ditentukan dari masukan pada waktu t saja, tetapi juga masukan pada waktu $t-1$, $t-2$, dst. Keunggulan utama dari RNNs adalah kemampuan untuk mengingat masukan sebelumnya yang dapat mempengaruhi luaran yang dihasilkan [20].

Meskipun, dalam teori, RNNs mampu untuk menyimpan informasi sekuensial dari data, dalam praktiknya, RNNs gagal karena masalah *vanishing* dan *exploding gradient* [21, 22] sehingga RNNs memiliki kelemahan apabila *data set* yang digunakan cukup besar [23]. LSTM [24] merupakan varian dari RNNs yang dirancang untuk mengatasi masalah *vanishing* dan *exploding gradient*.

2.2.2. Long Short-Term Memory

Pada dasarnya, LSTM terdiri dari tiga gerbang perkalian (*multiplicative gates*) yang mengontrol jumlah informasi yang akan dihilangkan, diteruskan, dan luaran langkah waktu berikutnya ($t+1$). Gambar 2.3 berikut ini adalah ilustrasi sederhana dari LSTM.



Gambar 2.3. LSTM unit [25]

Hak Cipta milik Universitas Pertamina
Universitas Pertamina
1. Dilarang mengutip karya tulis sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina:
a. menyebutkan sumber sesuai kaidah kepenulisan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
Dilarang mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
© Copyright of Universitas Pertamina

Fitur utama dari LSTM adalah adanya sebuah jalur yang menghubungkan *cell state* sebelumnya (C_{t-1}) ke *cell state* waktu sekarang (C_t). Dengan adanya jalur tersebut, suatu nilai di c_{t-1} dengan mudah diteruskan ke c_t dengan sedikit sekali modifikasi, sehingga dapat menghindari terjadinya *vanishing* dan *exploding gradient*.

Fitur lain dari LSTM adalah adanya gerbang *sigmoid* yang mengatur seberapa banyak informasi yang dapat dilewatkan dari data masukan. Untuk suatu masukan vektor x, luaran dari gerbang *sigmoid* adalah $\sigma(A \cdot x + b)$, dimana A adalah nilai bobot, b adalah bias, keduanya dipelajari selama proses pelatihan, dan σ adalah fungsi *sigmoid*. Luaran gerbang *sigmoid* tersebut digunakan untuk mengontrol seberapa banyak informasi dari C_{t-1} yang akan diteruskan ke C_t. Luaran gerbang *sigmoid* tersebut adalah angka antara 0 dan 1; luaran 0 artinya tidak ada informasi yang dilewatkan, sedangkan 1 artinya seluruh informasi dilewatkan.

Gerbang *sigmoid* pertama dari kiri yang ditunjukkan pada Gambar 2.3 disebut dengan *forget gate* (f_t). Pada gerbang *sigmoid* tersebut akan diputuskan informasi apa yang akan dilupakan dari masukan C_{t-1}. Masukan dari gerbang tersebut adalah nilai s_{t-1}dan x_t, dan menghasilkan nilai angka antara 0 dan 1 untuk setiap elemen dalam C_{t-1}.

Gerbang selanjutnya adalah gerbang yang digunakan untuk mengontrol informasi baru apa yang digunakan di C_t yang disebut dengan *input gate* (i_t). Gerbang sigmoid i_t memutuskan nilai mana yang akan diperbarui, kemudian sebuah lapisan $tanh$ menghasilkan kandidat vektor *cell state* baru $\tilde{C}_t$ yang selanjutnya digabungkan untuk membuat pembaruan ke *cell state* C_t.

Cell state yang baru (C_t) diperoleh dengan mengalikan *cell state* lama (C_{t-1}) dengan nilai *forget gate* (f_t) dan kemudian ditambahkan dengan hasil perkalian $\tilde{C}_t$ dan *input gate* (i_t).

Gerbang terakhir adalah *output gate* (o_t) yang merupakan gerbang *sigmoid* paling kanan pada Gambar 2.3 untuk memutuskan bagian-bagian C_t yang akan dihasilkan. Kemudian, nilai C_t dilewatkan pada fungsi aktivasi $tanh$ untuk membuat nilainya menjadi antara -1 dan 1, dan dikalikan dengan o_t sehingga hanya dihasilkan bagian yang telah diputuskan.

Secara umum, formula matematis untuk memperbarui sebuah *unit* LSTM pada waktu t ditulis dalam persamaan 2.2 – 2.7 sebagai berikut [25]:

$$f_t = \sigma(W_f \cdot [s_{t-1,} x_t] + b_f) \tag{2.2}$$

$$i_t = \sigma(W_i \cdot [s_{t-1,} x_t] + b_i) \tag{2.3}$$

$$\tilde{c}_t = tanh(W_c \cdot [s_{t-1,} x_t] + b_c) \tag{2.4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{2.5}$$

$$o_t = \sigma(W_o \cdot [s_{t-1,} x_t] + b_o) \tag{2.6}$$

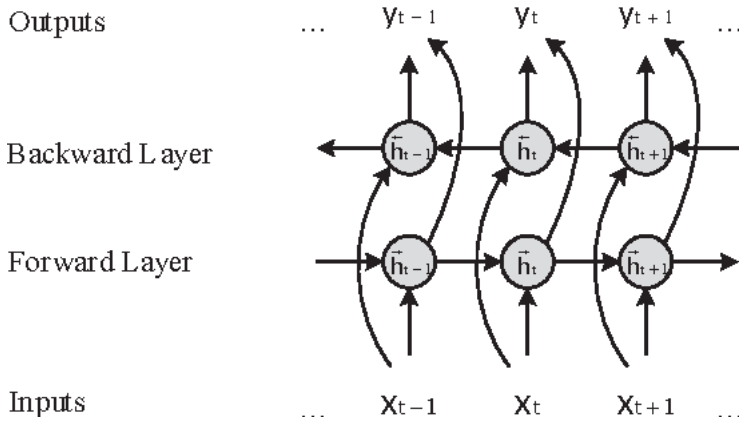
$$h_t = o_t \odot tanh\,(c_t) \tag{2.7}$$

dimana notasi $[s_{t-1,} x_t]$ dan $\odot$ merupakan operasi konkatenasi dan perkalian produk.

Untuk pelabelan sekuensial seperti NER dan *speech recognition*, LSTM sangat bermanfaat apabila ada akses untuk mengetahui informasi waktu sebelum dan setelahnya [26]. Namun, vektor luaran pada LSTM (s_t) hanya menerima informasi dari waktu sebelumnya saja, tidak punya akses informasi untuk waktu setelahnya. Solusi permasalahan tersebut terselesaikan berkat ditemukannya model

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
1. Dilarang mengutipkayatulis ini, sebagian atau seluruh karya tulis ini dalam bentuk apapun
a. mengutipan untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah atau penelitian;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
© Copyright of Universitas Pertamina

LSTM yang memiliki akses informasi dua arah yang disebut dengan *bi-directional* LSTM (BiLSTM) [27]. Arsitektur model *bidirectional* LSTM ditunjukkan pada Gambar 2.4.



Gambar 2.4. Bi-directional RNNs [28]

Ide dasarnya adalah menggunakan *forward* dan *backward* LSTM untuk menangkap informasi waktu sebelum dan setelahnya. Kemudian luaran dari *forward* dan *backward* LSTM tersebut digabungkan untuk memperoleh nilai luaran akhir.

2.2.3. Convolutional Neural Networks

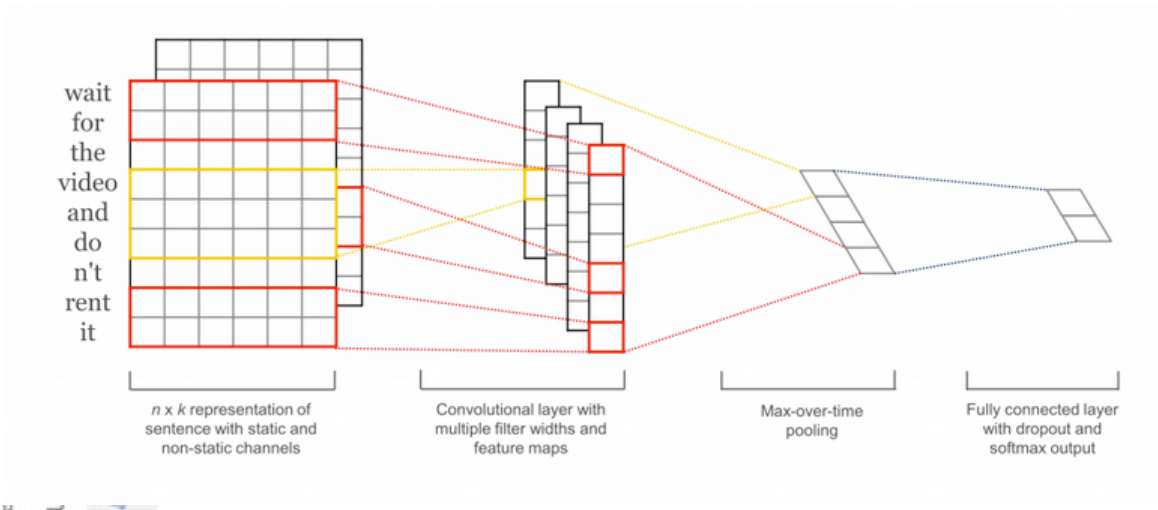
Convolutional Neural Networks (CNNs) merupakan salah satu jenis dari model NNs yang umumnya digunakan pada gambar. CNNs terdiri dari beberapa jenis lapisan, yaitu lapisan konvolusi, *pooling*, dan *fully connected*. Lapisan konvolusi digunakan untuk memperoleh luaran baru dari hasil perkalian dengan nilai setiap *filter*. Lapisan *pooling* digunakan untuk melakukan *sub-sampling* pada data luaran dari proses konvolusi dengan hanya mengambil fitur dominan saja. Di lapisan terakhir biasanya digunakaan *cross entropy loss* dengan fungsi *softmax* sebagai *classifier* [37]. *Cross entropy loss* dengan fungsi *softmax* atau *softmax loss* ditulis dalam persamaan 2.8 sebagai berikut.

$$CE = -log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right) \tag{2.8}$$

dimana s_p adalah skor hasil prediksi untuk kelas positif dan $\sum_j^C e^{s_j}$ jumlah seluruh skor kelas C.

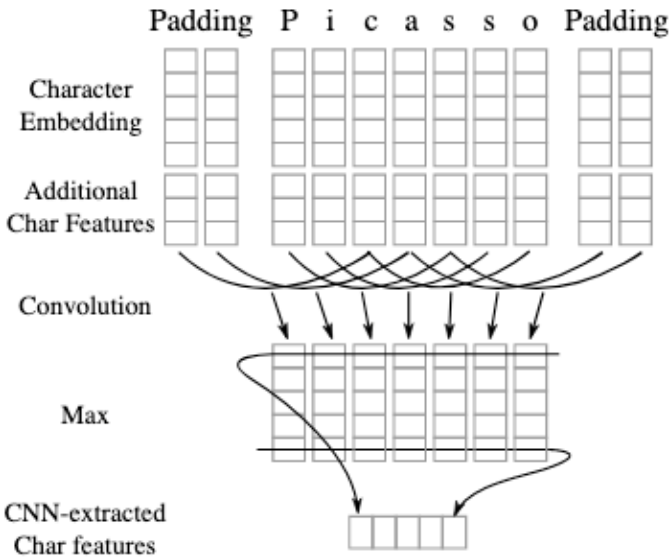
Meskipun umumnya digunakan pada gambar, tetapi beberapa penelitian telah menggunakan CNNs pada data teks. Penelitian [29] menggunakan arsitektur CNNs pada beberapa *data set* untuk melakukan klasifikasi. Arsitektur CNNs yang digunakan menunjukkan performa yang sangat bagus, meskipun arsitektur yang digunakan cukup sederhana dengan hanya menggunakan word2vec *embedding* di lapisan pertama, kemudian diikuti oleh lapisan konvolusi dengan beberapa *filters*, lapisan *max pooling*, dan terakhir lapisan *fully connected* dengan fungsi aktivasi *softmax*. Arsitektur model tersebut seperti yang ditunjukkan pada Gambar 2.5 berikut ini.

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
Dilarang mengutip karya tulis ini, sebagian atau seluruh:
a. menyebutkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
Dilindungi Undang-Undang
© Copyright Universitas Pertamina



Gambar 2.5. Lapisan CNNs untuk klasifikasi kalimat [29]

Berbeda dengan penelitian [29], penelitian [30] menggunakan CNNs sebagai *character-level embedding* untuk mengekstrak fitur dari setiap kata. Gambar 2.6 adalah arsitektur CNNs digunakan pada [30].



Gambar 2.6. Lapisan CNNs untuk character-level embedding [30]

Luaran dari lapisan *character-embedding* dan tipe dari karakter tersebut digunakan sebagai fitur tambahan dan kemudian dilakukan proses konvolusi untuk memperoleh representasi baru dari data dan fitur dominan dari kumpulan karakter dengan menggunakan *max pooling*.

2.3. Word Embedding

Seperti halnya dengan data gambar, data teks sebelum diolah dengan model ML harus terlebih dahulu diubah ke dalam bentuk angka. Data gambar dengan mudah diubah ke dalam angka karena memiliki nilai warna tertentu. Untuk data teks, proses pengubahan ke dalam bentuk vektor angka disebut dengan *embedding*. Hasil *embedding* tersebut digunakan dalam pemodelan ML dan berfungsi sebagai dasar untuk komputasi NLP.

Universitas Pertamina
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
ng mengumumkan karya tulis ini, kecuali:
menyebutkan sumber sesuai kaidah kecendekiaan;
Dilindungi Undang-Undang
© Copyright of Universitas Pertamina

Word-embedding menjadi alternatif pilihan dibandingkan dengan *language models* yang lain seperti Bag-of-Words (BoW) dan TF-IDF karena memiliki dimensi yang lebih rendah. Selain itu, *word embedding* juga dapat memperoleh nilai hubungan antar kata yang memiliki arti yang sama atau sering digunakan bersama.

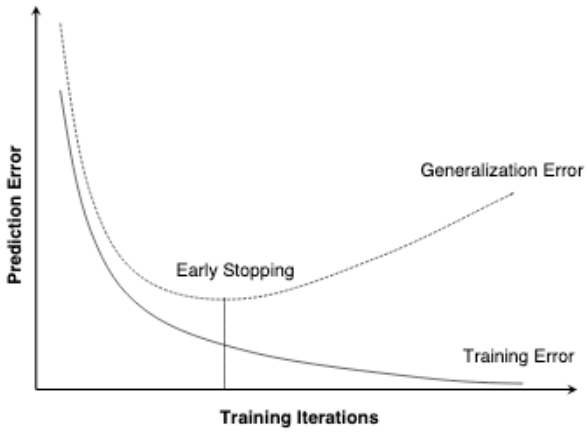
Penelitian [31] merupakan awal dari pengembangan *word-embedding* dengan membangun model untuk mengurangi dimensi data dengan mempelajari distribusi dari kata-kata tersebut. Setelah itu, penelitian terus dilakukan hingga menghasilkan teknik *embedding* dengan performa yang lebih baik [32, 33, 34]. Penelitian [32, 33] menciptakan word2vec dan [34] menciptakan GloVe yang masih banyak digunakan saat ini.

2.4. One-Hot-Encoding

One-hot-encoding digunakan dalam tahap pra-pemrosesan *data set* dengan jumlah kelas lebih dari dua. Metode ini menghasilkan vektor dengan panjang yang sama dengan jumlah kelas dalam *data set*. Jika sebuah observasi data termasuk ke dalam kelas i (indeks dari kelas), maka nilai elemen vektor tersebut diberi nilai 1 dan sisanya 0. Hal ini dilakukan untuk menghindari bias prediksi model apabila label dari kelas tersebut dianggap sebagai tipe data nominal atau ordinal dengan angka yang lebih tinggi memiliki bobot yang lebih tinggi.

2.5. Learning Curve

Learning curve merupakan sebuah ukuran performa model untuk melakukan prediksi terhadap jumlah upaya dan/atau pengalaman yang dilakukan seperti waktu, jumlah data, dan iterasi [38].



Gambar 2.7. *Learning curve* pada proses pelatihan *artificial neural network* [38]

Contoh *learning curve* pada Gambar 2.7 menunjukkan perubahan nilai *training error* dan *generalization error* terhadap jumlah iterasi selama proses pelatihan model. Terlihat bahwa nilai *generalization error* mengalami penurunan sampai akhirnya naik kembali, sedangkan nilai *training error* terus mengalami penurunan. Peningkatan nilai *generalization error* tersebut dikenal dengan istilah *overfitting*. Salah satu solusi untuk mengatasi masalah tersebut adalah dengan menggunakan *early stopping* untuk menghentikan proses pelatihan model apabila *generalization error* mengalami peningkatan dan/atau stagnan dalam beberapa iterasi.

Universitas Pertamina
Hak Cipta Milik Universitas Pertamina
Dilindungi Undang-Undang
© Copyright Universitas Pertamina
2. Dilarang mengutip sebagian atau seluruh karya tulis ini, kecuali:
a. mencantumkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

2.6. Metrik Evaluasi

Performa model untuk melakukan klasifikasi dapat diukur dengan menghitung jumlah kelas yang diprediksi dengan benar (*true positive*), jumlah diprediksi bukan termasuk kelas tersebut dan benar (*true negative*), dan yang salah prediksi (*false positive* atau *false negative*). Gambar 2.8 adalah *confusion matrix* yang dapat digunakan untuk mengidentifikasi jumlah *true positive*, *true negative*, *false positive*, dan *false negative*.

Data class	Classified as *pos*	Classified as *neg*
pos	true positive (tp)	false negative (fn)
neg	false positive (fp)	true negative (tn)

Gambar 2.8. *Confusion-matrix* untuk klasifikasi biner [35]

Terdapat beberapa metrik klasifikasi yang dapat digunakan, yaitu akurasi, *precision*, *recall*, dan *fscore*. Nilai metrik tersebut dapat dihitung dari *confusion matrix* dengan persamaan 2.8 – 2.11 sebagai berikut [35]:

$$Accuracy = \frac{tp + tn}{tp + fn + fp + tn} \tag{2.8}$$

$$Precision = \frac{tp}{tp + fp} \tag{2.9}$$

$$Recall = \frac{tp}{tp + fn} \tag{2.10}$$

$$fscore = \frac{(\beta^2 + 1)tp}{(\beta^2 + 1)\ tp + \beta^2 fn + fp} \tag{2.11}$$

Metrik *precision*, *recall*, dan *fscore* digunakan untuk mengukur performa model dalam melakukan prediksi pada kelas *positive*.

Untuk klasifikasi *multi-class*, dua pendekatan yang dapat digunakan untuk memperoleh nilai rataan metrik, yaitu dengan rataan makro dan mikro. Rataan makro menghitung metrik secara independen setiap kelas dan kemudian dibagi dengan jumlah kelas untuk memperoleh nilai metrik rata-rata, sedangkan rataan mikro menggabungkan kontribusi semua kelas untuk menghitung metrik rata-rata. Dengan kata lain, rataan makro memperlakukan setiap kelas setara, sedangkan rataan mikro tetap mempertimbangkan jumlah sampel pada setiap kelas. Nilai rataan makro dan mikro tersebut dapat dihitung dari *confusion matrix* dengan persamaan 2.12 – 2.17 sebagai berikut [35]:

$$Micro\ precision = \frac{\sum_{i=1}^{m} tp_i}{\sum_{i=1}^{m}(tp_i + fp_i)} \tag{2.12}$$

$$Micro\ recall = \frac{\sum_{i=1}^{m} tp_i}{\sum_{i=1}^{m}(tp_i + fn_i)} \tag{2.13}$$

Universitas Pertamina
Hak Cipta Milik Universitas Pertamina
Dilindungi Undang-Undang
© Copyright Universitas Pertamina
1. Dilarang mengutipkarya tulis ini, kecuali:
a. menyebutkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

$$Micro\ fscore = \frac{(\beta^2 + 1)Micro\ precision\ \cdot Micro\ recall}{\beta^2 Micro\ precision + Micro\ recal} \tag{2.14}$$

$$Macro\ precision = \frac{\sum_{i=1}^{m} \frac{tp_i}{tp_i + fp_i}}{m} \tag{2.15}$$

$$Macro\ recall = \frac{\sum_{i=1}^{m} \frac{tp_i}{tp_i + fn_i}}{m} \tag{2.16}$$

$$Macro\ fscore = \frac{(\beta^2 + 1)Macro\ precision\ \cdot Macro\ recall}{\beta^2 Macro\ precision + Macro\ recal} \tag{2.17}$$

dimana m adalah jumlah kelas; i adalah indeks kelas ke i; dan β adalah parameter bobot *precision* dan *recall* terhadap *nilai fscore*.

Nilai β yang biasa digunakan adalah 1,0 dan disebut dengan *f1 score*. Pada *f1 score*, bobot *precision* dan *recall* sama dalam perhitungan *fscore*. Nilai β yang lebih kecil, seperti 0,5, memberikan bobot lebih besar pada *precision* dibanding *recall*, sedangkan nilai β yang lebih besar, seperti 2,0, memberikan bobot yang lebih kecil pada *precision* dibanding *recall*.

2.7. Penelitian NER Terdahulu

Penelitian terkait NER dalam bahasa Indonesia telah dilakukan oleh beberapa peneliti sebelumnya. Beberapa penelitian tersebut adalah sebagai berikut:

1. Agus Willyawan melakukan penelitian terkait NER dalam bahasa Indonesia dengan menggunakan *Conditional Random Field* dan *Pos-Tagging*. Penelitian tersebut hanya membatasi NEs pada nama orang, tempat, dan lokasi berdasarkan teks yang dimasukkan. *Data set* yang digunakan dari Wikipedia dan DBpedia. Performa model yang dihasilkan pada penelitian tersebut memperoleh nilai *precision* dan *recall* di atas 75% dan *f1 score* sebesar 77% [36].
2. D. W. Wulandari et. al. melakukan penelitian terkait NER pada dokumen biologi menggunakan metode *rule-based* dan *naïve bayes classifier*. Penelitian tersebut menggunakan 19 dokumen latih yang diproses dan dianotasi manual untuk mencari NEs dan didapat 1.135 data latih dalam bentuk kata. Pra-pemrosesan data yang dilakukan diantaranya adalah *pos-tagging* dan *n-gram*. Dari kombinasi metode *rule-based* dan *naïve bayes*, penelitian tersebut memperoleh hasil rata-rata *precision*, *recall*, dan *f-measure* sebesar 0.8 dengan *micro average* [15].
3. W. Gunawan et. al. melakukan penelitian terkait NER dalam bahasa Indonesia menggunakan metode *Bidirectional* LSTM-CNNs. Lapisan CNNs digunakan untuk membangun representasi *multi-sequence* pada model menggunakan lima lapisan konvolusi satu dimensi dengan ukuran kernel 2, 4, 6, 8, dan 10. Kernel dengan ukuran yang berbeda tersebut sama dengan representasi *n-gram* dimana *n* adalah ukuran kernel pada lapisan konvolusi satu dimensi. Penelitian tersebut fokus pada bahasa Indonesia untuk mengekstrak informasi 4 kelas NEs, yaitu *Person*, *Organization*, *Location*, dan *Event*. *Data set* yang digunakan sebanyak 4.139 kalimat dan 81.173 kata yang diperoleh dari artikel berita dan artikel tentang sejarah Indonesia di Wikipedia. Penelitian tersebut membandingkan beberapa pendekatan model DL untuk memperoleh hasil yang optimal. Dari beberapa model yang dibandingkan, model BLSTM-CNNs memiliki performa yang paling tinggi dengan nilai *precision*, *recall*, dan *f1-score* sekitar 79% [6].

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
Dilarang mengutip karya tulis ini, kutipan:
a. dengan mencantumkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

Dari ketiga penelitian di atas, yang akan dilakukan pada penelitian ini mirip dengan yang dilakukan oleh [6] menggunakan model BiLSTM-CNNs. Perbedaan penelitian [6] dan yang dilakukan pada penelitian ini adalah sebagai berikut.

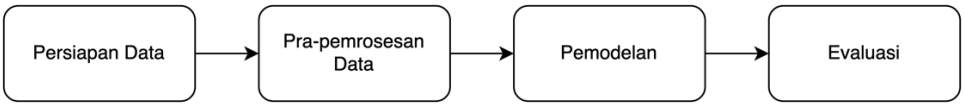
1. NEs yang digunakan pada penelitian ini adalah Person, Organization, Location, Quantity, dan Time, tetapi dengan format yang sama menggunakan format BILOU (*beginning*, *inside*, *last*, *outside*, *unit*). Dengan format pelabelan BILOU, luaran model NER yang diperoleh tidak hanya memprediksi jenis NEs, tetapi posisi token pada NEs dapat dijelaskan dengan baik, sehingga konteks dari NEs yang diperoleh lebih detail.
2. Dari beberapa model yang diuji pada [6], model BLSTM-CNNs-LSTM merupakan model yang paling kompleks, tetapi hasil yang diperoleh paling rendah karena jumlah *data set* yang digunakan tidak dapat mengakomodasi kompleksitas model untuk belajar. Oleh karena itu, pada penelitian ini, digunakan arsitektur model yang lebih sederhana lagi dibanding model BiLSTM-CNNs yang memiliki performa tertinggi di penelitian tersebut. Model BiLSTM-CNNs yang dibangun hanya menggunakan satu lapisan konvolusi dan satu jenis ukuran kernel saja.

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
Dilindungi Undang-Undang
© Copyright of Universitas Pertamina
1. Dilarang mengutip karya tulis ini, kecuali:
a. menyebutkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

Universitas
Pertamina

BAB III
METODE PENELITIAN

Bab ini menguraikan metodologi penelitian yang telah dilakukan. Sub-bab 3.1 menjelaskan tentang pengumpulan dan persiapan data yang digunakan. Pada sub-bab 3.2 diuraikan tentang metode pra-pemrosesan data yang dilakukan. Sub-bab 3.3 dan 3.4 dijelaskan tentang metode pemodelan dan evaluasi hasil prediksi model. Gambar 3.1 berikut ini menunjukkan alur kerja penelitian secara keseluruhan.



Gambar 3.1. Alur kerja penelitian

3.1. Persiapan Data

Data set yang digunakan pada penelitian ini adalah data berupa kumpulan kalimat yang telah diberi label dan diperoleh dari sumber *data set* publik dari dua repositori di github, yakni repositori nlp-experiments[1] dan indonesian-ner[2]. *Data set* tersebut menggunakan format pelabelan MUC-6 [13] dan tag XML. Format pelabelan *data set* tersebut diubah dari format MUC-6 [13] dan tag XML menjadi format BILOU. BILOU merupakan singkatan dari ***B**eginning*, ***I**nside*, dan ***L**ast* token untuk NEs *multi-token*, serta ***O**utside* dan ***U**nit* untuk NEs *single-token*. Format pelabelan BILOU secara signifikan mengungguli BIO yang telah digunakan secara luas [14].

3.2. Pra-pemrosesan Data

Data dalam bentuk kumpulan tag NEs yang diperoleh kemudian diolah agar data tersebut dapat diproses dengan model (sub-bab 3.3). Tujuan utama pada tahap ini adalah mengubah data ke dalam bentuk pelabelan sekuensial. Langkah-langkah yang dilakukan adalah sebagai berikut.

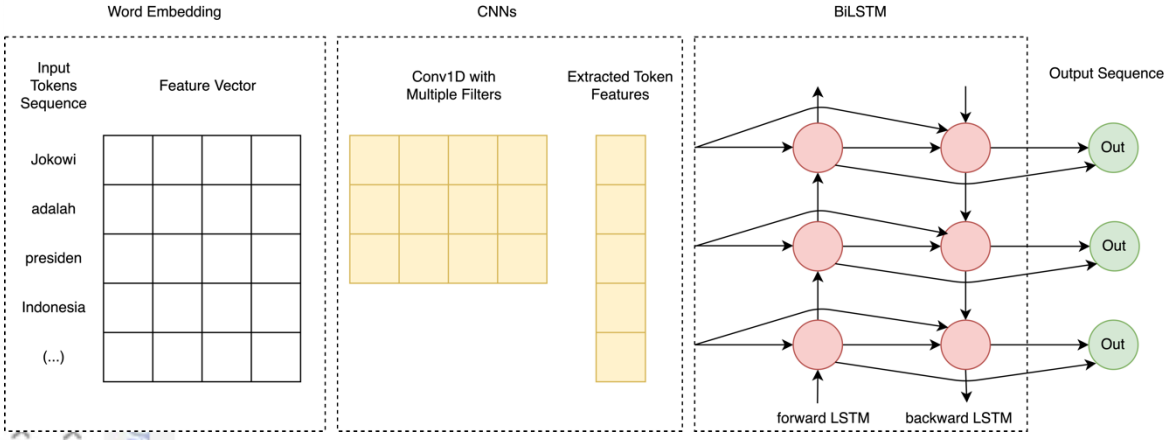
1. Seluruh token dan label diberi indeks.
2. Menggabungkan setiap token beserta labelnya ke dalam sebuah *sequence* sama yang akan digunakan sebagai observasi data. Misal kalimat "Jokowi adalah presiden Indonesia." Diubah menjadi [[("Jokowi", "U-PERSON"), ("adalah", "O"), ("presiden", "O"), ("Indonesia", "U-LOCATION"), (".", "O"), (...)], ...]
3. Token dan label tersebut kemudian dipisah untuk membentuk data masukan X dan luaran y.
4. Setiap observasi data masukan X harus memiliki panjang *sequence* yang sama sehingga ditambahkan *padding* dengan kata "PADDING" dan diberi indeks terakhir. Label *padding* yang ditambahkan adalah O.
5. Luaran y kemudian diubah ke dalam bentuk *one-hot-encoding*.
6. Dimensi X dan y masing-masing adalah (jumlah seluruh observasi data, panjang *sequence*) dan (jumlah observasi data, panjang sequence, jumlah kelas).
7. Langkah terakhir adalah membagi data ke dalam data latih, data validasi, dan data uji.

[1] https://github.com/yohanesgultom/nlp-experiments
[2] https://github.com/yusufsyaifudin/indonesia-ner

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
Dilindungi Undang-Undang Copyright © Universitas Pertamina
1. Dilarang menggunakan seluruh atau sebagian karya tulis ini, kecuali
a. untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

3.3. Pemodelan

Model yang diuji pada penelitian ini terdiri dari tiga jenis model, yaitu BiLSTM, BiLSTM-CNNs, dan BiLSTM-CNNs + multi-sequence n-gram [6] dengan menggunakan word2vec dan *trainable embedding* dari keras. Model BiLSTM dipilih karena telah digunakan di beberapa penelitian terdahulu untuk pelabelan *sequence* dengan performa yang baik [6, 26, 30]. Dibandingkan dengan [6, 30], penelitian ini hanya menggunakan satu lapisan konvolusi dan satu jenis ukuran kernel dan tidak menggunakan *character-level embedding* untuk menjaga agar model tetap sederhana dengan waktu pemrosesan yang rendah. Model CNNs digunakan karena menghasilkan performa yang baik dalam klasifikasi teks di beberapa penelitian dengan arsitektur model yang cukup sederhana [29]. Model yang dibangun setidaknya terdiri dari 3 bagian utama seperti yang ditunjukkan pada Gambar 3.2 berikut ini.



Gambar 3.2. Arsitektur model BiLSTM-CNNs yang digunakan

Masukan pada model adalah adalah *sequence X* dengan dimensi sesuai dengan ukuran *mini batch* setiap iterasi dan panjang *sequence*. *Padding* pada *sequence* tersebut bukan *padding* yang ditambahkan pada lapisan CNNs, melainkan *padding* yang ditambahkan saat pra-pemrosesan data agar seluruh observasi data memiliki panjang *sequence* yang sama.

Lapisan pertama pada model tersebut adalah lapisan *word-level embedding* yang digunakan untuk mengubah masukan *sequence* ke dalam bentuk vektor berdimensi-n. *Word-level embedding* yang digunakan adalah word2vec yang dilatih dengan menggunakan pustaka gensim[3] dan *data set* dari wikipedia bahasa Indonesia[4]. Setiap baris pada vektor tersebut merupakan representasi dari setiap token dengan panjang *n*. Luaran dari lapisan *word-level embbeding* adalah vektor dengan dimensi yang sama dengan panjang *sequence* x *n* yang merupakan representasi baru dari masukan *sequence*.

Luaran pada lapisan *embbeding* selanjutnya digunakan sebagai masukan pada lapisan Conv1D untuk mengekstrak informasi dengan melakukan proses konvolusi. Lapisan Conv1D melakukan konvolusi dengan menggerakkan sebuah kernel konvolusi (*filter*) dengan ukuran tertentu. Luaran dari lapisan Conv1D adalah informasi representatif baru dari masukan *sequence* dari hasil perkalian lapisan *embedding* dengan nilai setiap *filter* yang digunakan. Pada lapisan CNNs ini tidak menggunakan lapisan *pooling* untuk menjaga dimensi dan informasi *sequence* dari data tetap sama.

[3] https://radimrehurek.com/gensim/index.html
[4] https://dumps.wikimedia.org/idwiki/latest/

Universitas Pertamina
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.

menentukan luaran NEs dari data. *Output sequence* yang merupakan label setiap token hasil prediksi model diperoleh dari penambahan luaran *forward* dan *backward* LSTM.

Selama proses pelatihan digunakan *softmax loss* atau *categorical cross entropy* sebagai *loss function* pada model. Pelatihan dilakukan dengan maksimal jumlah *epoch* tertentu, tetapi dihentikan apabila selama beberapa *epoch* nilai *loss function* pada model tidak mengalami penurunan. Hal tersebut dilakukan untuk menghindari *overfit* pada model terhadap data latih.

Seluruh model yang dibangun pada penelitian ini menggunakan bahasa pemrograman Python dan pustaka Tensorflow[5] dengan antarmuka Keras[6]. Proses optimasi akan dilakukan untuk setiap model dengan melakukan hiper-parameter *tuning* dengan mencari kombinasi nilai hiper-parameter yang menghasilkan performa paling tinggi. Proses hiper-parameter *tuning* menggunakan bantuan pustaka talos[7].

3.4. Evaluasi

Performa model diukur dengan menggunakan metrik akurasi *f1 score* selama proses pelatihan dan pengujian untuk setiap label. Metrik *f1 score* adalah metrik *fscore* (sub-bab 2.5) dengan nilai $\beta = 1$. Performa setiap model secara keseluruhan diukur menggunakan rataan mikro. Pemilihan rataan mikro pada penelitian ini didasarkan pada distribusi label pada data yang digunakan tidak seimbang (Tabel 4.1), sehingga metrik *f1 score* yang diperoleh tetap mempertimbangkan distribusi setiap kelas. Semakin tinggi nilai metrik evaluasi baik untuk setiap label maupun nilai rataan secara keseluruhan berarti semakin tinggi performa model tersebut. Label O (termasuk *padding*) dihilangkan pada saat evaluasi performa model.

[5] https://www.tensorflow.org/
[6] https://keras.io/
[7] https://github.com/autonomio/talos

Universitas Pertamina
Hak Cipta milik Universitas Pertamina © Copyright of Universitas Pertamina
Dilarang mengutipkarya tulis ini kecuali: Dilarang mengutipkarya tulis ini kecuali:
a. mengebukan untuk seluruh kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

Universitas
Pertamina

BAB IV
HASIL DAN PEMBAHASAN

Pada bab ini dibahas hasil penelitian yang telah dilakukan. Sub-bab 4.1 dan 4.2 menjelaskan tentang hasil yang diperoleh pada tahap persiapan dan pra-pemrosesan data. Pada sub-bab 4.3 akan diuraikan proses yang dilakukan selama pemodelan ML beserta hasil dan analisis terhadap hasil yang diperoleh. Sub-bab 4.4 akan diuraikan hasil analisis performa setiap model dari sub-bab sebelumnya.

4.1. Persiapan Data

Dari gabungan dua sumber *data set* yang digunakan diperoleh total 4.892 kalimat, 98.122 kata 13.031 kata unik, dan 5 jenis NEs, yaitu nama orang, organisasi, lokasi, kuantitas, dan waktu. Untuk mengubah kedua *data set* tersebut ke dalam format BILOU dilakukan proses *parsing* yang berbeda. *Data set* nlp-experiments yang menggunakan format pelabelan MUC-6 terlebih dahulu dipisah per kalimat (akhir kalimat ditandai dengan tanda baca titik). Kemudian data yang diapit oleh tag <ENAMEX type="<NE TYPE>"></ENAME> diekstrak dan diberi label sesuai nilai *type* di *tag* tersebut. Token yang muncul dua kali secara berurutan dengan tipe *tag* yang sama diberi label B-<NE TYPE> untuk token pertama dan L-<NE TYPE> untuk token kedua. Apabila token tersebut muncul lebih dari dua kali secara berurutan dengan tipe *tag* yang sama, maka perlakuannya sama dengan dua token, tetapi token ke 2 sampai token sebelum token terakhir diberi label I-<NE TYPE>. Untuk token yang hanya muncul satu kali dan tidak diapit oleh *tag* diberi label masing-masing U-<NE TYPE> dan O. Proses *parsing* untuk *data set* indonesian-ner kurang lebih sama, yang membedakan adalah *tag* pada *data set* tersebut menggunakan tipe NEs, sehingga proses *parsing* menggunakan nama *tag* langsung untuk menentukan label setiap token.

Dari proses *parsing* tersebut diperoleh 21 jenis label dalam format BILOU, dimana yang terbanyak adalah label O. Sedangkan label yang termasuk dalam NEs paling banyak adalah U-PERSON dengan jumlah 1.861 dan yang paling sedikit adalah label U-QUANTITY yang hanya berjumlah 6 token. Jumlah kemunculan untuk setiap label dapat dilihat pada Tabel 4.1 berikut ini.

Tabel 4.1. Jumlah kemunculan label

Label	Jumlah	(%)
O	81787	83,35%
U-PERSON	1861	1,89%
U-ORGANIZATION	1722	1,75%
U-LOCATION	1502	1,53%
B-PERSON	1467	1,49%
L-PERSON	1464	1,49%
L-ORGANIZATION	1036	1,05%
B-ORGANIZATION	1035	1,05%
I-ORGANIZATION	941	0,95%
I-TIME	865	0,88%
L-LOCATION	728	0,74%
B-LOCATION	726	0,73%
I-LOCATION	488	0,49%
L-QUANTITY	474	0,48%

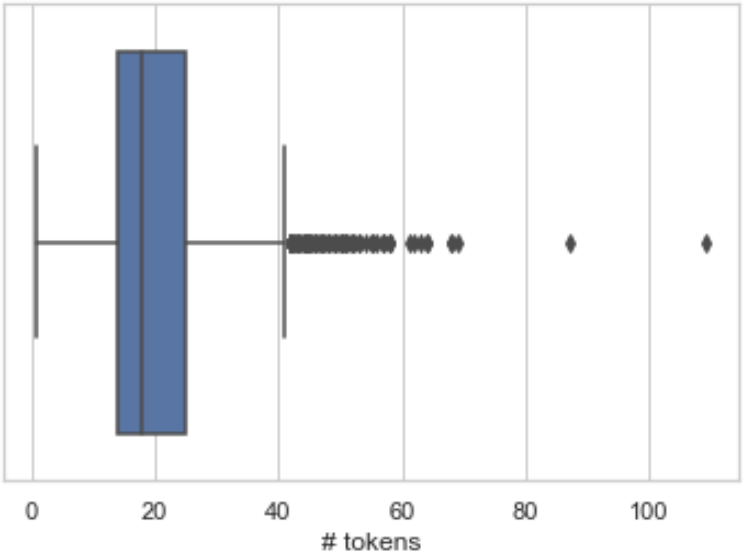
Universitas Pertamina
Hak Cipta milik Universitas Pertamina
1. Dilarang mengutip karya tulis ini, secara
a. mengutipnya hanya untuk kepentingan pendidikan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun
tanpa adanya izin dari Universitas Pertamina.
© Copyright of Universitas Pertamina

Label	Jumlah	(%)
B-QUANTITY	474	0,48%
I-PERSON	470	0,47%
I-QUANTITY	363	0,36%
B-TIME	282	0,28%
L-TIME	282	0,28%
U-TIME	149	0,15%
U-QUANTITY	6	0,006%

Data set yang akan digunakan sangat tidak seimbang jumlah antar labelnya dimana yang paling banyak adalah token label O dengan 83,35% dari *data set*.

4.2. Pra-pemrosesan Data

Panjang *sequence* pada *data set* ditentukan berdasarkan pertimbangan jumlah token pada setiap observasi data. Distribusi jumlah token untuk setiap observasi data ditunjukkan pada Gambar 4.1 berikut ini.



Gambar 4.1. Distribusi panjang token setiap observasi data

Jumlah token pada *data set* paling banyak berada pada rentang 14 (quartile 1) sampai dengan 25 (quartile 3) token, serta yang paling panjang adalah 109 token. Berdasarkan distribusi jumlah token tersebut, jumlah token yang dipilih untuk diuji pada tahap pemodelan adalah 20, 40, 60, dan 80 token. Pemilihan jumlah token yang digunakan sebagai panjang *sequence* pada data harus dilakukan dengan baik untuk menghindari penambahan *noise* terlalu banyak dan/atau kehilangan informasi akibat dari pemilihan panjang *sequence* yang kurang tepat. Jumlah data latih, data validasi, dan data uji masing-masing adalah 70%, 15%, dan 15% dari *data set*.

Hak Cipta milik Universitas Pertamina
Dilindungi Undang-Undang
© Copyright Universitas Pertamina
Universitas Pertamina
1. Dilarang mengutip karya tulis ini, kecuali:
a. menyebutkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

4.3. Pemodelan

Terdapat tiga jenis arsitektur model yang diuji, yaitu model BiLSTM, BiLSTM-CNNs, dan BiLSTM-CNNs + multi-sequence n-gram dan terdapat dua pendekatan *word-embedding* yang diuji, yaitu *trainable* dan word2vec *embedding*. Dua model tersebut dilatih dengan menggunakan nilai hiper-parameter awal seperti yang ditunjukkan pada Tabel 4.2 berikut ini. Nilai hiper-parameter setiap model akan dioptimasi pada sub sub-bab 4.3.5.

Tabel 4.2. Nilai hiper-parameter awal pada model

Hiper-parameter	Nilai Awal
Max epoch	100
Loss function	softmax loss
Metric	f1 score micro average
Early stopping	monitor = val_f1_score_micro; patience = 10
Batch size	128
Learning algorithm	Adam; learning rate = 0,001
Dropout	0,1
LSTM units	100
CNNs filters	128; size = 3

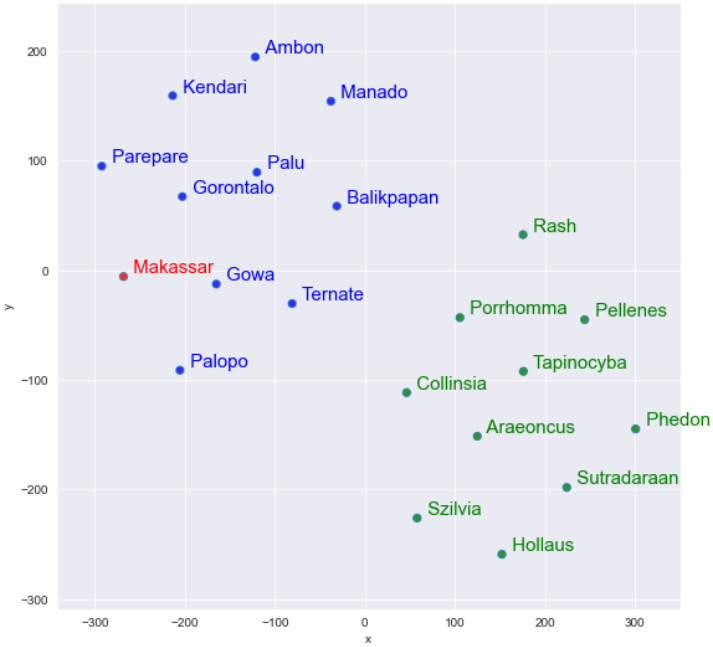
4.3.1. Embedding

Lapisan pertama pada model yang digunakan pada penelitian ini seperti yang ditunjukkan pada Gambar 3.3 adalah lapisan *embedding*. Ada dua jenis metode *embedding* yang diuji, yaitu *trainable* dan word2vec *embedding*. Kedua metode tersebut memiliki fungsi yang sama untuk mengubah data teks (kumpulan kata atau token) ke dalam representasi angka yang menyimpan hubungan semantik antar kata.

Cara kerja kedua metode *embedding* tersebut berbeda. *Trainable embedding* yang disediakan oleh pustaka Keras merupakan *supervised learning* dengan nilai bobot yang diinisiasi secara acak dan kemudian diperbaharui selama proses pelatihan model dengan menggunakan algoritma *back-propagation*. Sedangkan word2vec *embedding* merupakan jenis *unsupervised learning* yang berupaya mencari kata yang sering muncul dalam konteks serupa dan diletakkan berdekatan satu sama lain dalam *embedding space* [32].

Pada penelitian ini, word2vec dilatih dengan menggunakan 406.068 artikel yang terdiri 496.380.181 kata dari wikipedia bahasa Indonesia dengan waktu pelatihan selama 702 detik. Pendekatan yang digunakan dalam proses pelatihan adalah menggunakan metode skip-gram dan *negative sampling* dengan nilai parameter *window size* 5, *negative samples* 5, dan dimensi luaran 400. Luaran dari pelatihan tersebut adalah sebuah model *pretrained* word2vec *embedding* yang digunakan pada lapisan *embedding* setiap model. Contoh penggunaan *pretrained* word2vec *embedding* tersebut adalah mencari kata yang mirip dengan kata tertentu berdasarkan konteks pemakaian seperti yang ditunjukkan pada Gambar 4.2.

Universitas Pertamina
Hak cipta milik Universitas Pertamina
Dilindungi Undang-Undang
1. Dilarang mengutip karya tulis ini, kecuali:
a. menyebutkan sumber sesuai dengan kaidah keseddkian;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.



Gambar 4.2. Kata-kata yang mirip dengan "Makassar": (biru) kata yang sangat mirip dan (hijau) yang sangat tidak mirip

Kata "Makassar" memiliki arti yang serupa dengan "Gowa", "Palopo", "Parepare", dan kata lainnya. Dari seluruh kata yang diprediksi mirip dengan kata "Makassar" tersebut, semuanya merupakan nama lokasi atau tempat di Indonesia bagian timur. Representasi angka setiap kata tersebut digunakan sebagai bobot pada lapisan *embedding* tanpa diperbaharui selama proses pelatihan. Salah satu kelebihan dari penggunaan *pretrained embedding* adalah dapat mengurangi jumlah *trainable parameter* pada model.

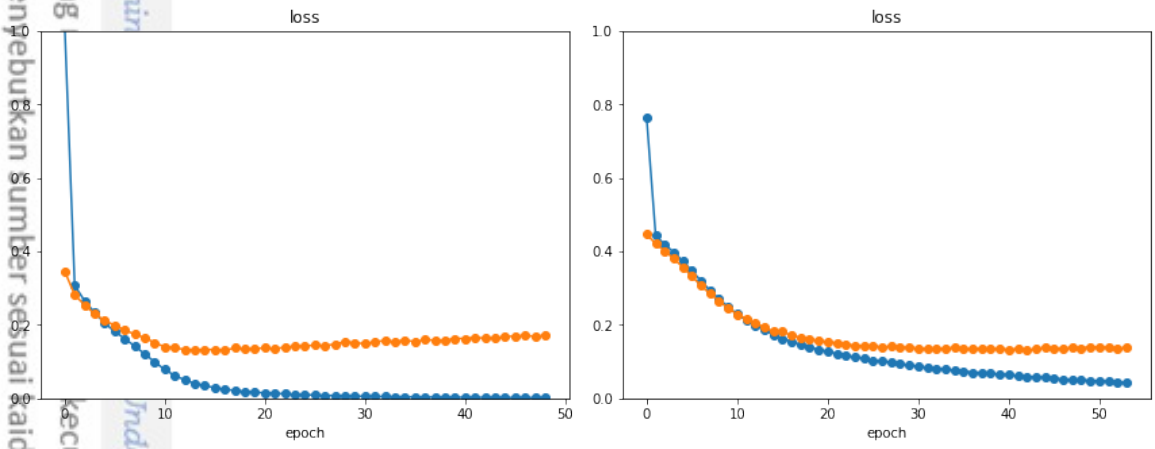
Hak Cipta milik Universitas Pertamina
Dilindungi Undang
© Copyright of Universitas Pertamina
Universitas Pertamina
1. Dilarang mengutip karya tulis ini kecuali
a. menyebutkan sumber sesuai dengan pecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

4.3.2. BiLSTM

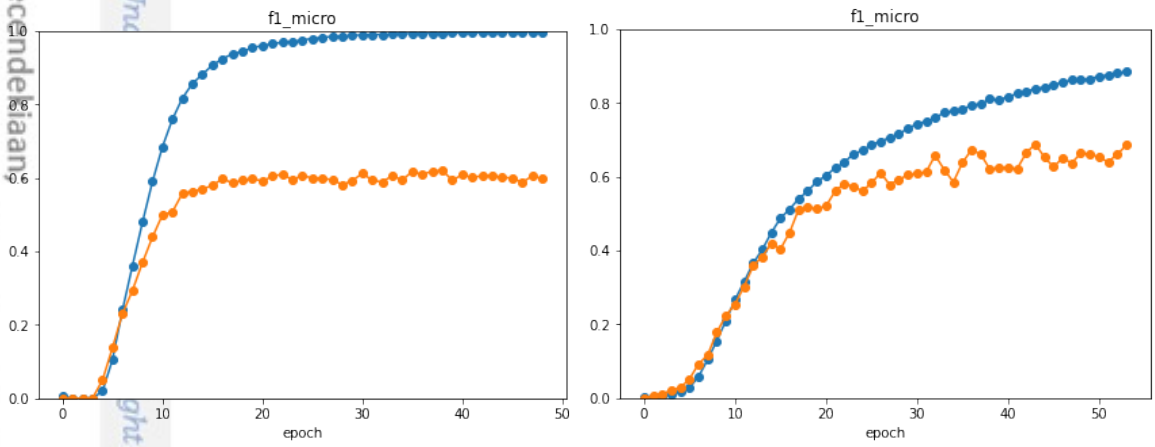
Gambar 4.3 berikut ini adalah arsitektur model yang digunakan pada model BiLSTM.

Gambar 4.3. Arsitektur model BiLSTM

Pelatihan model menggunakan empat jenis panjang *sequence*, yaitu 20, 40, 60, dan 80 seperti yang telah dijelaskan pada sub-bab 3.2. Gambar 4.4 dan 4.5 berikut ini adalah *learning curve* proses pelatihan model yang memiliki panjang *sequence* paling optimal sesuai dengan perolehan nilai *f1 score* pada Tabel 4.3.



Gambar 4.4. Perubahan nilai *loss*: *trainable* (kiri) dan word2vec (kanan)



Gambar 4.5. Perubahan nilai *f1 score*: *trainable* (kiri) dan word2vec (kanan)

Model yang menggunakan *trainable embedding* mengalami *overfit* terhadap data latih, sedangkan word2vec cenderung lebih tahan terhadap *overfit* yang terlihat dari perbedaan nilai *loss* dan *f1 score* pelatihan dan validasi. Hal tersebut besar kemungkinan disebabkan oleh jumlah *learning parameters* pada model dengan *trainable embedding* jauh lebih tinggi daripada model yang menggunakan word2vec *embedding*. Dengan arsitektur dan nilai parameter yang sama, model yang menggunakan word2vec *embedding* hanya memiliki 405.021 *learning parameters,* sedangkan model yang menggunakan *trainable embedding* sebanyak 5.617.421.

Universitas Pertamina
Hak Cipta Milik Universitas Pertamina
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber sesuai dengan ketentuan yang berlaku di Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
a. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber sesuai dengan ketentuan yang berlaku di Universitas Pertamina, kecuali:
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.

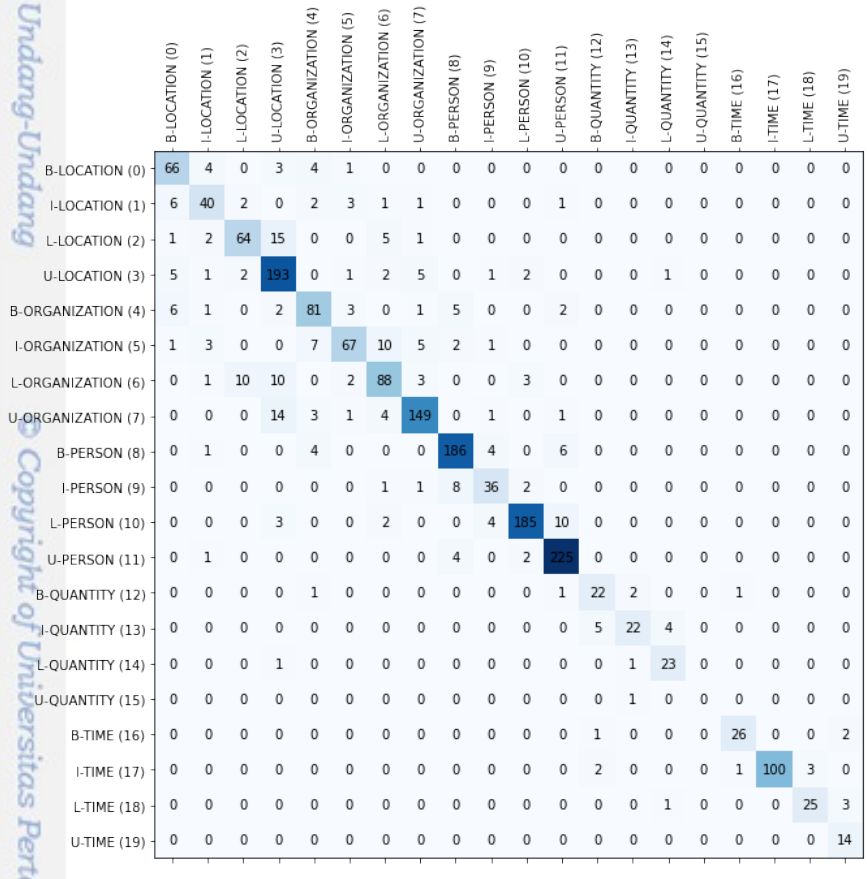
Tabel 4.3 berikut ini adalah hasil evaluasi model menggunakan data uji dan nilai hiper-parameter awal setiap panjang *sequence* yang berbeda.

Tabel 4.3. Nilai *f1 score* setiap pemilihan panjang *sequence* pada model BiLSTM

Panjang Sequence	BiLSTM + trainable emb.		BiLSTM + word2vec	
	# Training Time / Epoch	F1 Score	# Training Time / Epoch	F1 Score
20	118 detik / 30	59,66%	73 detik / 77	62,59%
40	123 detik / 29	60,22%	61 detik / 54	**67,70%**
60	281 detik / 49	**62,11%**	124 detik / 61	64,30%
80	183 detik / 40	60,42%	80 detik / 64	62,82%

Model yang menggunakan word2vec *embedding* memiliki nilai *f1 score* lebih tinggi daripada model yang menggunakan *trainable embedding* untuk semua skema pemilihan panjang *sequence* dan dengan *training time* yang jauh lebih rendah. Rata-rata perbedaan nilai *f1 score* antara word2vec dan *trainable embedding* adalah 3,75%. Panjang *sequence* yang memiliki nilai *f1 score* paling tinggi adalah 40 *tokens* dengan nilai 67,70%.

Gambar 4.6 berikut ini adalah *confusion matrix* hasil evaluasi menggunakan data uji pada model dengan perolehan nilai *f1 score* tertinggi. Beberapa kesalahan prediksi terjadi pada entitas dan label yang sama, seperti I-ORGANIZATION diprediksi 10 kali sebagai U-ORGANIZATION dan L-ORGANIZATION diprediksi 14 kali sebagai U-LOCATION.

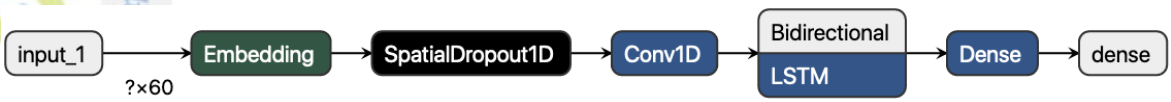


Gambar 4.6. *Confusion matrix* pada data uji pada model BiLSTM

Universitas Pertamina
Hak cipta milik Universitas Pertamina (Pertamina University)
Dilarang mengutip kata/tulisan, kecuali:
a. mencantumkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
Dilarang mengutip sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Universitas Pertamina.
Hak Cipta milik Universitas Pertamina (Pertamina University)
Dilindungi Undang-Undang
Copyright of Universitas Pertamina

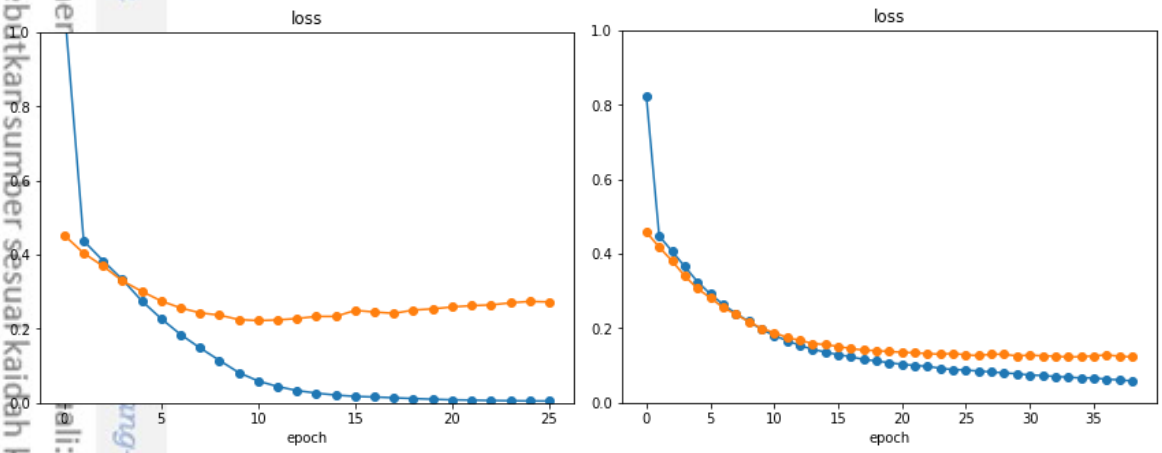
4.3.3. BiLSTM-CNNs

Gambar 4.7 berikut ini adalah arsitektur model yang digunakan pada model BiLSTM-CNNs. Perbedaan dengan model yang pertama (sub sub-bab 4.3.2) adalah adanya lapisan Conv1D pada model ini yang digunakan untuk melakukan proses konvolusi pada nilai *embedding* setiap *sequence*.

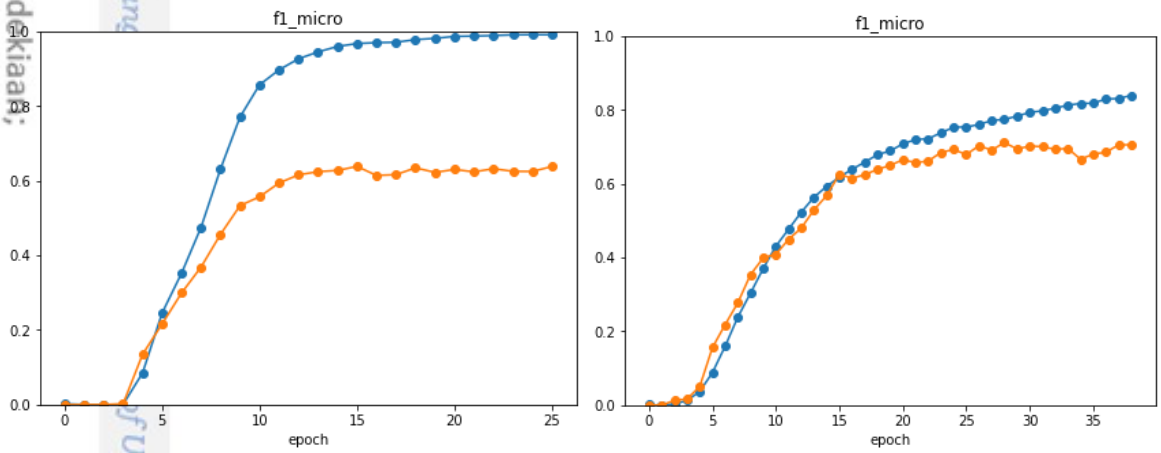


Gambar 4.7. Arsitektur model BiLSTM-CNNs

Pelatihan model tetap menggunakan empat jenis panjang *sequence* dan nilai parameter yang sama dengan model sebelumnya. Gambar 4.8 dan 4.9 berikut ini adalah *learning curve* selama proses pelatihan model. *Learning curve* yang ditampilkan hanya untuk panjang *sequence* yang memiliki nilai *f1 score* paling tinggi.



Gambar 4.8. Perubahan nilai *loss*: *trainable* (kiri) dan word2vec (kanan)



Gambar 4.9. Perubahan nilai *f1 score*: *trainable* (kiri) dan word2vec (kanan)

Dengan jumlah *learning parameters* yang jauh lebih rendah, model dengan word2vec *embedding* lebih tahan terhadap *overfit* daripada model yang menggunakan *trainable embedding*. Hal tersebut besar kemungkinan disebabkan oleh jumlah *learning parameters* pada model dengan *trainable embedding* jauh lebih tinggi daripada model yang menggunakan word2vec *embedding*. Model yang menggunakan word2vec hanya memiliki 341.149 *learning parameters*, sedangkan *trainable*

Universitas Pertamina
Hak Cipta dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber sesuai kaidah kecendekiaan;
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
a. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
b. pengutipan tidak merugikan Universitas Pertamina.

embedding memiliki 5.553.549. Jumlah *learning parameters* pada BiLSTM-CNNs + word2vec bahkan lebih rendah dari model BiLSTM + word2vec.

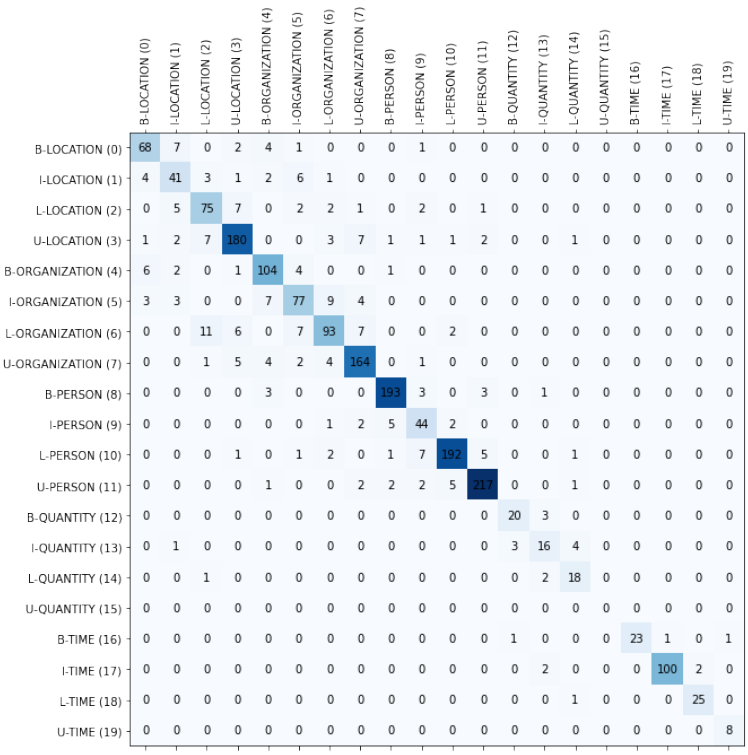
Tabel 4.4 berikut ini adalah hasil evaluasi model menggunakan data uji dan nilai hiper-parameter awal setiap panjang *sequence* yang berbeda.

Tabel 4.4. Nilai *f1 score* setiap pemilihan panjang *sequence* pada model BiLSTM-CNNs

Panjang Sequence	**BiLSTM-CNNs + trainable emb.**		**BiLSTM-CNNs + word2vec**	
	# Training Time / Epoch	**F1 Score**	**# Training Time / Epoch**	**F1 Score**
20	162 detik / 41	60,37%	45,4 detik / 45	68,95%
40	110 detik / 26	**62,11%**	46,1 detik / 39	**69,63%**
60	138 detik / 24	61,98%	100 detik / 46	68,07%
80	120 detik / 27	61,13%	77 detik / 60	69,51%

Model yang menggunakan word2vec *embedding* memiliki nilai *f1 score* lebih tinggi daripada model yang menggunakan *trainable embedding* untuk semua skema pemilihan panjang *sequence* dan dengan *training time* yang jauh lebih rendah. Rata-rata perbedaan nilai *f1 score* antara word2vec dan trainable embedding adalah 7,64%. Nilai *f1 score* tertinggi diperoleh oleh model yang menggunakan word2vec *embedding* pada panjang *sequence* 40 *tokens*.

Gambar 4.10 berikut ini adalah *confusion matrix* hasil evaluasi menggunakan data uji pada model dengan perolehan nilai *f1 score* tertinggi. Dibandingkan dengan model pertama (Gambar 4.8), kesalahan prediksi lebih dari 10 pada model ini hanya terjadi pada label yang sama, yaitu L-ORGANIZATION diprediksi 11 kali sebagai L-LOCATION.

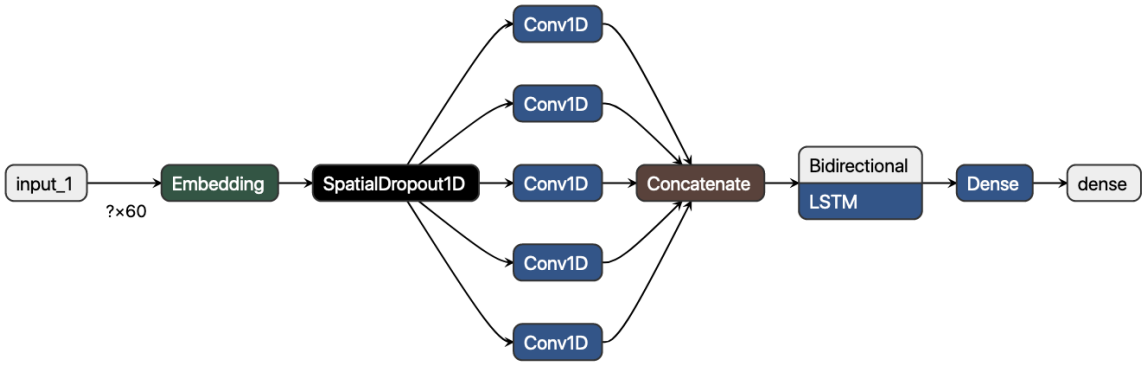


Gambar 4.10. *Confusion matrix* pada data uji pada model BiLSTM-CNNs

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
Dilarang menutup sebagian atau seluruh karya tulis ini dengan cara/alat apapun
a. menyebutkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
© Copyright of Universitas Pertamina

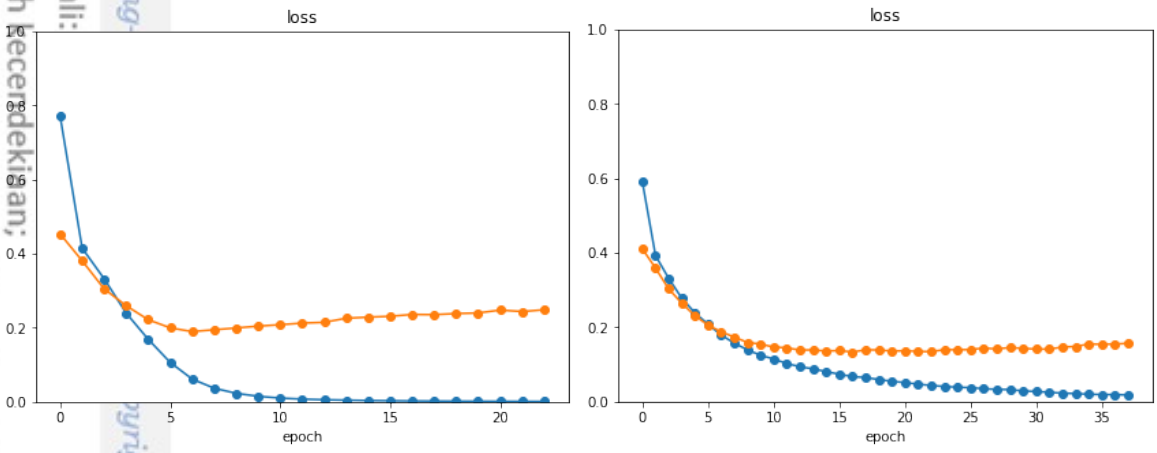
4.3.4. BiLSTM-CNNs + multi-sequence n-gram

Gambar 4.11 berikut ini adalah arsitektur model yang digunakan pada [6]. Perbedaan dengan model yang kedua (sub sub-bab 4.3.3) adalah adanya beberapa lapisan Conv1D dengan ukuran *kernel* 2, 4, 6, 8, dan 10 pada model ini. Ukuran *kernel* yang berbeda tersebut digunakan untuk membentuk luaran *multi-sequence* dari lapisan CNNs. Luaran *multi-sequence* tersebut dalam bentuk n-gram hasil proses konvolusi dengan *n* adalah ukuran *kernel* yang digunakan di setiap lapisan Conv1D.



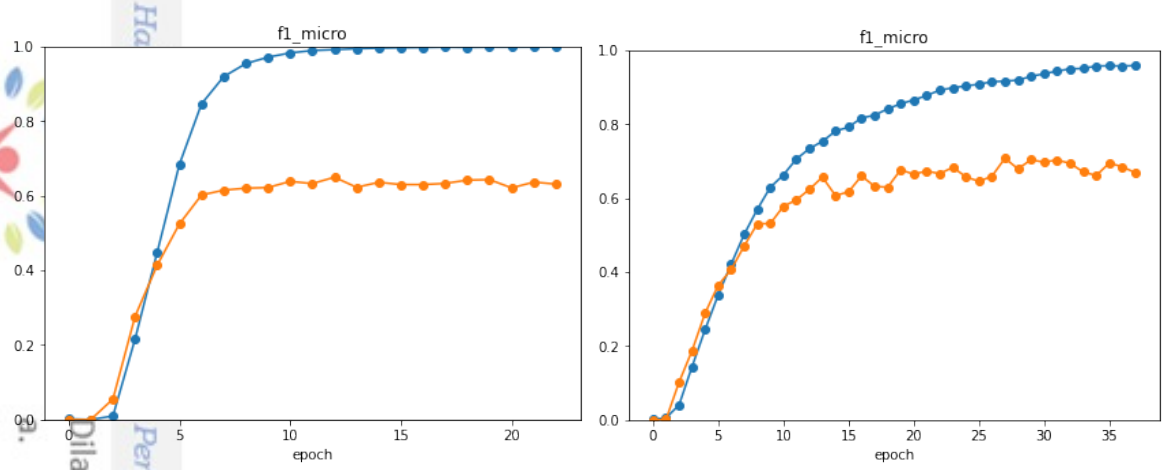
Gambar 4.11. Arsitektur model BiLSTM-CNNs + multi-sequence n-gram

Pelatihan model tetap menggunakan empat jenis panjang *sequence*, tetapi nilai parameter berbeda dengan sebelumnya. Model ini memiliki *dropout rate* sebesar 0,5 dan empat lapisan Conv1D dengan ukuran kernel yang berbeda. Gambar 4.12 dan 4.13 berikut ini adalah *learning curve* selama proses pelatihan model. *Learning curve* yang ditampilkan hanya untuk panjang *sequence* yang memiliki nilai *f1 score* paling tinggi.



Gambar 4.12. Perubahan nilai *loss*: *trainable* (kiri) dan word2vec (kanan)

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
Dilarang mengutip sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Universitas Pertamina.
Copyright of Universitas Pertamina
karya tulis ini, sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Universitas Pertamina;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.



Gambar 4.13. Perubahan nilai *f1 score*: *trainable* (kiri) dan word2vec (kanan)

Tren pada dua model sebelumnya juga terjadi pada model ini. Model dengan word2vec *embedding* lebih tahan terhadap *overfitting* daripada model yang menggunakan *trainable embedding*. Model yang menggunakan word2vec hanya memiliki 2.133.661 *learning parameters*, sedangkan *trainable embedding* memiliki 7.346.061.

Tabel 4.5 berikut ini adalah hasil evaluasi model menggunakan data uji dan nilai hiper-parameter awal setiap panjang *sequence* yang berbeda.

Tabel 4.5. Nilai *f1 score* setiap pemilihan panjang *sequence* pada model BiLSTM-CNNs + multi-sequence n-gram

Panjang Sequence	**BiLSTM-CNNs + multi-sequence n-gram + trainable emb.**		**BiLSTM-CNNs + multi-sequence n-gram + word2vec**	
	# Training Time / Epoch	**F1 Score**	**# Training Time / Epoch**	**F1 Score**
20	99 / 22	**65,27%**	48,6 / 41	64,61%
40	112 / 23	64,13%	64 / 38	**66,06%**
60	274 / 39	64,30%	180 / 55	65,67%
80	111 / 22	64,06%	99 / 46	65,51%

Model yang menggunakan word2vec *embedding* memiliki nilai *f1 score* rata-rata 1,02% lebih tinggi daripada model yang menggunakan *trainable embedding*, tetapi perbedaan tersebut relatif lebih rendah dibanding dua model sebelumnya. Panjang *sequence* yang memiliki nilai *f1 score* paling tinggi adalah 40 *tokens* dengan nilai 66,06%.

Gambar 4.14 berikut ini adalah *confusion matrix* hasil evaluasi menggunakan data uji pada model dengan perolehan nilai *f1 score* tertinggi. Kesalahan prediksi dengan jumlah lebih besar atau sama dengan 10 hanya terjadi pada entitas I-ORGANIZATION yang diprediksi 10 kali sebagai B-ORGANIZATION dan L-ORGANIZATION.

Universitas Pertamina
Hak Cipta Dilindungi Undang-Undang © Copyright Universitas Pertamina
Dilarang mengutip kayatus ini, kecuali:
1. menyebutkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

	B-LOCATION (0)	I-LOCATION (1)	L-LOCATION (2)	U-LOCATION (3)	B-ORGANIZATION (4)	I-ORGANIZATION (5)	L-ORGANIZATION (6)	U-ORGANIZATION (7)	B-PERSON (8)	I-PERSON (9)	L-PERSON (10)	U-PERSON (11)	B-QUANTITY (12)	I-QUANTITY (13)	L-QUANTITY (14)	U-QUANTITY (15)	B-TIME (16)	I-TIME (17)	L-TIME (18)	U-TIME (19)
B-LOCATION (0)	65	6	0	5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I-LOCATION (1)	6	38	3	0	3	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0
L-LOCATION (2)	0	4	73	8	0	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0
U-LOCATION (3)	1	2	8	184	0	1	1	9	0	1	1	2	0	0	0	0	0	0	0	0
B-ORGANIZATION (4)	4	2	0	2	83	4	0	1	1	0	0	0	0	0	0	0	0	0	0	0
I-ORGANIZATION (5)	3	3	0	0	10	51	10	6	1	1	1	0	0	0	0	0	0	0	0	0
L-ORGANIZATION (6)	0	0	9	8	0	0	89	5	0	1	3	0	0	0	0	0	0	0	0	0
U-ORGANIZATION (7)	0	0	1	6	3	0	3	161	0	0	2	3	0	0	0	0	0	0	0	0
B-PERSON (8)	1	1	0	0	2	0	0	0	180	3	0	1	0	0	0	0	0	0	0	0
I-PERSON (9)	0	0	0	0	0	0	0	1	8	42	2	0	0	0	0	0	0	0	0	0
L-PERSON (10)	0	0	2	1	0	0	1	0	0	4	191	4	0	0	0	0	0	0	0	0
U-PERSON (11)	0	0	0	0	1	0	0	0	4	0	8	210	0	0	0	0	0	0	0	0
B-QUANTITY (12)	0	1	0	0	0	0	0	0	0	0	0	1	16	1	0	0	0	0	0	0
I-QUANTITY (13)	0	0	0	0	0	0	0	1	0	0	0	0	4	17	2	0	1	0	0	0
L-QUANTITY (14)	0	0	0	1	0	0	0	0	0	0	0	0	0	1	19	0	0	0	0	0
U-QUANTITY (15)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
B-TIME (16)	0	0	1	1	0	0	0	0	0	0	0	0	2	0	0	0	23	1	0	2
I-TIME (17)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	95	2	0
L-TIME (18)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	26	0
U-TIME (19)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	10

Gambar 4.14. *Confusion matrix* pada data uji pada model BiLSTM-CNNs + multi-sequence n-gram + word2vec

Hak Cipta milik Universitas Pertamina Dilindungi Undang-Undang © Copyright of Universitas Pertamina

Universitas Pertamina

1. Dilarang mengutip karya tulis ini, kecuali:
   a. menyebutkan sumber sesuai kaidah kecendekiaan;
   b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
   c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

4.3.5. Optimisasi Hiper-parameter

Berdasarkan hasil pelatihan model dengan nilai parameter awal yang ditunjukkan pada tiga sub sub-bab sebelumnya, diperoleh bahwa model yang menggunakan word2vec *embedding* memiliki performa yang jauh lebih baik daripada *trainable embedding*. Hasil tersebut diperoleh karena setiap vektor representasi kemiripan token telah didefinisikan pada *pretrained* word2vec *embedding*, sedangkan bobot pada *trainable embedding* harus diperbaharui berdasarkan nilai *loss function* selama proses *back-propagation*. Selain itu, panjang *sequence* yang paling optimal adalah 40 *tokens* untuk semua model. Jika kita lihat pada Gambar 4.1, nilai 40 terletak di sekitar nilai maksimal sebelum nilai *outliers* (*max whisker*). Dari dua hasil tersebut diputuskan bahwa optimasi nilai hiper-parameter hanya dilakukan pada word2vec *embedding* dan dengan panjang *sequence* 40 *tokens*.

Optimasi hiper-parameter dilakukan dengan mencari nilai kombinasi hiper-parameter baru dari nilai awal pada Tabel 4.2. Pemilihan nilai uji merupakan bagian terpenting pada proses optimasi hiper-parameter agar parameter terbaik termasuk dalam daftar nilai uji yang didefinisikan. Terdapat empat parameter yang akan dioptimasi, yaitu *learning algorithm* (*optimizer*) & *learning rate*, jumlah LSTM *units*, jumlah CNNs *filters* & *filter size*, dan nilai *dropout rate*. Pemilihan nilai uji untuk *learning algorithm* didasarkan pada [40] yang menunjukkan bahwa *learning algorithm* Nadam, Adam, dan RMSprop memperoleh nilai *f1 score* tertinggi dibanding *optimizer lainnya*. Nilai uji hiper-parameter yang lain dipilih dengan mengambil beberapa nilai di bawah dan di atas nilai awal pada Tabel 4.2. Misalnya, nilai awal parameter LSTM *units* adalah 100, maka nilai uji yang dipilih adalah [50;100; 200]. Pengecualian untuk nilai *dropout rate* yang hanya dipilih nilai yang lebih besar dari nilai awal karena model cenderung mengalami *overfit* berdasarkan *learning curve* yang ditunjukkan pada Gambar 4.4, 4.5, 4.8, 4.9, 4.12, dan 4.13. Hasil optimasi hiper-parameter untuk kedua model tersebut ditunjukkan pada Tabel 4.6 – 4.8 berikut ini.

Tabel 4.6. Optimasi nilai hiper-parameter pada model BiLSTM + word2vec

Round	Hiper-parameter	Nilai Uji	Nilai Terbaik	F1 Score
1	Learning algorithm	[Nadam; Adam; RMSprop]	Adam	72,23%
	Learning rate	[0,01; 0,001; 0,0001]	0,01	
2	LSTM units	[50; 100; 200]	200	72,17%
3	Dropout	[0,1; 0,3; 0,5; 0,7]	0,7	74,67%

Tabel 4.7. Optimasi nilai hiper-parameter pada model BiLSTM-CNNs + word2vec

Round	Hiper-parameter	Nilai Uji	Nilai Terbaik	F1 Score
1	Learning algorithm	[Nadam; Adam; RMSprop]	Adam	73,78%
	Learning rate	[0,01; 0,001; 0,0001]	0,01	
2	LSTM units	[50; 100; 200]	50	71,70%
3	CNNs filters	[32; 64; 128; 256]	64	74,57%
	Filter size	[2; 3; 4]	3	
4	Dropout	[0,1; 0,3; 0,5; 0,7]	0,7	78,34%

Universitas Pertamina
1. Dilarang mengutip bagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
a. Mengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
b. pengutipan tidak merugikan Universitas Pertamina.
c. Dilarang mengumumkan dan memperbanyak karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
© Copyright by Universitas Pertamina

Tabel 4.8. Optimasi nilai hiper-parameter pada model BiLSTM-CNNs + multi-sequence n-gram + word2vec

Round	Hiper-parameter	Nilai Uji	Nilai Terbaik	F1 Score
1	Learning algorithm	[Nadam; Adam; RMSprop]	Adam	74,94%
	Learning rate	[0,01; 0,001; 0,0001]	0,01	
2	LSTM units	[50; 100; 200]	100	74,38%
3	CNNs filters	[32; 64; 128; 256]	64	73,21%
4	Dropout	[0,1; 0,3; 0,5; 0,7]	0,7	76,34%

Pemilihan nilai terbaik tersebut didasarkan pada perolehan nilai *f1 score* validasi tertinggi berdasarkan nilai parameter uji.

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
Dilindungi Undang-Undang
© Copyright of Universitas Pertamina
1. Dilarang mengutip karya tulis ini, kecuali:
a. menyebutkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

4.4. Analisis

Setelah proses optimasi nilai hiper-parameter, model kembali dievaluasi dengan menggunakan data uji. Hasil evaluasi kedua model tersebut ditunjukkan pada Tabel 4.9 – 4.11 berikut ini.

Tabel 4.9. Perbandingan performa model pada data uji dengan nilai hiper-parameter terbaik

Model	# Trainable Params	Training Time / Epoch	F1 Score
BiLSTM + word2vec	405.021	21,3 detik / 19	70,44%
BiLSTM-CNNs + word2vec	213.085	36,6 detik / 32	**71,37%**
BiLSTM-CNNs + multi-sequence n-gram + word2vec	1.109.341	59,9 detik / 39	67,25%

Optimasi nilai hiper-parameter yang dilakukan pada sub sub-bab 4.3.5 dapat meningkatkan performa setiap model dari nilai parameter awal. Model BiLSTM + word2vec mengalami peningkatan performa sebesar 2,74% dari 67,70% menjadi 70,44% (Tabel 4.3), model BiLSTM-CNNs + word2vec mengalami peningkatan sebesar 1,44% dari 69,63% menjadi 71,37% (Tabel 4.4), dan model BiLSTM-CNNs + multi-sequence n-gram + word2vec mengalami peningkatan sebesar 1,19% dari 66,06% menjadi 67,25% (Tabel 4.5).

Tabel 4.10. Perbandingan performa model pada rata-rata nilai *f1 score* setiap NEs

NEs	% Kemunculan	BiLSTM + word2vec	BiLSTM-CNNs + word2vec	BiLSTM-CNNs + multi-sequence n-gram + word2vec
PERSON	5,36%	**82,27%**	78,76%	78,77%
ORGANIZATION	4,82%	63,32%	**65,24%**	61,78%
LOCATION	3,50%	72,29%	**74,50%**	68,67%
TIME	1,60%	**69,83%**	65,14%	57,62%
QUANTITY	1,34%	33,04%	**33,98%**	21,14%

Tabel 4.11. Perbandingan performa model pada rata-rata nilai *f1 score* setiap posisi token

Posisi Token	% Kemunculan	BiLSTM + word2vec	BiLSTM-CNNs + word2vec	BiLSTM-CNNs + multi-sequence n-gram + word2vec
B	4,06%	67,45%	**68,44%**	61,27%
I	3,18%	59,43%	**59,99%**	54,55%
L	4,06%	**70,32%**	70,16%	63,46%
O	83,35%	**98,62%**	98,51%	98,33%
U	5,34%	**59,40%**	55,50%	51,35%

Universitas Pertamina
Hak Cipta milik Universitas Pertamina (Dilindungi Undang-Undang)
© Copyright of Universitas Pertamina
1. Dilarang mengutip sebagian atau seluruh karya tulis ini, kecuali:
a. mencantumkan sumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

Hasil evaluasi dengan data uji menunjukkan bahwa model BiLSTM-CNNs + word2vec memiliki nilai *f1 score* tertinggi dengan jumlah *trainable parameters* yang lebih rendah. Hasil tersebut diperoleh karena pada model BiLSTM-CNNs terdapat lapisan Conv1D yang dapat mengekstrak informasi dari lapisan *embedding* melalui proses konvolusi menjadi sebuah representasi informasi baru dalam bentuk *feature map* yang digunakan sebagai masukan pada model BiLSTM.

Namun, model BiLSTM-CNNs + multi-sequence n-gram + word2vec dengan lapisan Conv1D yang lebih banyak justru memiliki performa yang rendah, bahkan jika dibandingkan dengan model BiLSTM yang tidak memiliki lapisan Conv1D. Hal tersebut kemungkinan disebabkan karena lapisan Conv1D yang banyak dan dengan ukuran filter yang besar menyebabkan model *overfitting* dan kehilangan informasi *sequence* dari data. Dugaan adanya kehilangan informasi *sequence* pada data akibat dari penggunaan ukuran *filter* yang besar didasarkan pada hasil optimasi hiper-parameter dengan ukuran *filter* terbaik adalah 3 (Tabel 4.7), sedangkan pada model BiLSTM-CNNs + multi-sequence n-gram + word2vec menggunakan ukuran *filter* 2, 4, 6, 8, dan 10. Hal tersebut juga ditunjukkan pada Tabel 4.10 dan 4.11 dimana model BiLSTM-CNNs + multi-sequence n-gram + word2vec tidak unggul di NEs dan posisi token apapun.

Selain memiliki nilai *f1 score* tertinggi, model BiLSTM-CNNs + word2vec juga cenderung dapat memprediksi setiap NEs dengan baik (Tabel 4.10) dan dapat membedakan letak token pada NEs *multi-token* (Tabel 4.11) dibanding model lain.

Hak Cipta Dilindungi Undang-Undang © Copyright of Universitas Pertamina
Universitas Pertamina
1. Dilarang mengutipkan tulis ini, kecuali:
a. merencukarsumber sesuai kaidah kecendekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

Universitas
Pertamina

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut:

- Dengan jumlah *trainable parameter* 70,95–93,85% lebih rendah, model dengan word2vec *embedding* memiliki performa 1,02%–3,75% lebih baik dibanding *trainable embedding* dilihat dari nilai *f1 score* rataan mikro. Hal ini disebabkan karena setiap vektor representasi kemiripan kata telah terdefinisikan pada *pretrained* word2vec *embedding* dan dilatih dengan jumlah data yang jauh lebih besar, sedangkan bobot pada *trainable embedding* harus diperbaharui berdasarkan nilai *loss function* selama proses *back-propagation*.
- Optimasi nilai hiper-parameter dapat meningkatkan performa setiap model dengan peningkatan sekitar 2,74% untuk model BiLSTM + word2vec, 1,44% untuk model BiLSTM-CNNs + word2vec, dan 1,19% untuk model BiLSTM-CNNs + multi-sequence n-gram + word2vec.
- Model BiLSTM-CNNs + word2vec dapat memprediksi 3 dari 5 NEs lebih bagus dan dapat membedakan letak token pada NEs *multi-token* dibanding model lain.
- Model terbaik untuk melakukan NER pada penelitian ini adalah model BiLSTM-CNNs + word2vec *embedding* dengan nilai *f1 score* rataan mikro sebesar 71,37%.

5.2. Saran

Untuk penelitian selanjutnya sebaiknya menggunakan metode yang dapat mengatasi *overfit* model terhadap data latih. Penggunaan *pretrained embedding* seperti word2vec dan teknik *early stopping* dapat mengurangi *overfitting* pada penelitian ini, tetapi perbedaan nilai antara *f1 score* validasi saat optimasi nilai hiper-parameter dan *f1 score* pada data uji masih sangat besar dengan perbedaan sekitar 7% untuk model BiLSTM-CNNs + word2vec.

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
© Copyright of Universitas Pertamina
1. Dilarang mengutip karya tulis ini, kecuali:
a. menyebutkan sumber sesuai kaidah kependekiaan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

Universitas
Pertamina

DAFTAR PUSTAKA

[1] R. Alfred, L. C. Leong, C. K. On, and P. Anthony, "Malay Named Entity Recognition Based on Rule-Based Approach," International Journal of Machine Learning and Computing, vol. 4, no. 3, pp. 300 – 306, 2014.

[2] D. Christianto, E. Siswanto, and R. Chaniago, "Penggunaan Named Entity Recognition dan Artificial Intelligence Markup Language untuk Penerapan Chatbot Berbasis Teks," Jurnal Telematika, vol. 10, no. 2, pp. 61 – 68, 2015.

[3] E. Adamopoulou and L. Moussiades, "An Overview of Chatbot Technology," IFIP International Conference on Artificial Intelligence Applications and Innovations, vol. 584 IFIP, pp. 373-383, 2020.

[4] S. Ayanouz, B. A. Abdelhakim, M. Benhmed, "A Smart Chatbot Architecture based NLP and Machine Learning for Health Care Assistance," in ACM International Conference Proceeding Series, pp. 1 – 6, 2020

[5] M. Y. S. Dirgantara, M. A. Fauzi, and R. S. Perdana, "Penerapan Named Entity Recognition Untuk Mengenali Fitur Produk Pada E-Commerce Menggunakan Rule Template Dan Hidden Markov Model," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 2, pp. 3912-3920, 2018.

[6] W. Gunawan, D. Suhartono, F. Purnomo, and A. Ongko, "Named-Entity Recognition for Indonesian Language using Bidirectional LSTM-CNNs," 3rd International Conference on Computer Science and Computational Intelligence, vol. 135, pp. 425-432, 2018.

[7] P. Sun, X. Yang, X. Zhao, and Z. Wang, "An Overview of Named Entity Recognition," In Proc. of 2018 International Conference on Asian Language Processing (IALP), 2018, Bandung, Indonesia [Online]. Available: IEEE Xplore, https://ieeexplore.ieee.org/document/8629225. [Accessed: 25 Agustus 2020].

[8] E. Utami and S. Hartati, Pendekatan Metode Rule Based Dalam Mengalihbahasakan Teks Bahasa Inggris Ke Teks Bahasa Indonesia, Jurnal Informatika, vol. 8, pp. 42-53, 2007.

[9] K. Shaalan, "Rule-based Approach in Arabic Natural Language Processing," vol. 3, no. 3, 2010.

[10] B. Bustami, "Penerapan Algoritma Naïve Bayes Untuk Mengklasifikasi Data Nasabah Asuransi," Jurnal Informatika, vol. 8, 2014.

[11] G. Nguyen, S. Dlugolinsky, M. Bobak, V. Tan, A. L, Garcia, I. Heredia, P. Malik, and L. Hluchy, "Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: survey," Aritificial Intelligence Review, vol 52, pp. 77-124, 2019.

[12] R. Shargnat, "Named Entity Recognition: A Literature Survey," 2014 [Online]. Available: http://www.cfilt.iitb.ac.in/resources/surveys/rahul-ner-survey.pdf. [Accessed: 26 Agustus 2020].

[13] R. Grishma and B. Sundheim, "Message Understanding Conference – 6: A Brief History," In Proc. of The 16th International Conference on Computational Linguistics, 1996 [Online]. Available: ACL Anthology, https://www.aclweb.org/anthology/C96-1079/. [Accessed: 26 Agustus 2020].

[14] L. Ratinov and D. Roth, "Design Challenges and Misconceptions in Named Entity Recognition," In Proc. of Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), Boulder, Colorado, pp. 147-155, 2009 [Online]. Available: https://www.aclweb.org/anthology/W09-1119/. [Accessed: 26 Agustus 2020].

Universitas Pertamina

Hak Cipta Universitas Pertamina, Dilindungi Undang-Undang Hak Cipta Universitas Pertamina

1. Dilarang mengutipkarya tulisan, kecuali:
a. menyebutkan sumber sesuai kaidah pencedekian;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

[15] D. W. Wulandari, P. P. Adikara, and S. Adinugroho, "Named Entity Recognition (NER) pada Dokumen Biologi Menggunakan Rule Based dan Naïve Bayes Classifier," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 2, no. 11, pp. 4555-4563, 2018.
[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[17] H. Helmiriawan, "Scalability Analysis of Predictive Maintenance Using Machine Learning in Oil Refineries", Master thesis, Delft Universite of Technology, 2018 [Online]. Available: http://resolver.tudelft.nl/uuid:dbf3e77c-e624-47ef-b951-3f1948b1609a. [Accessed: 26 Agustus 2020].
[18] R. Silipo, "Neural Networks", in Intelligence Data Analysis: An Introduction, Springer Berlin Heidelberg, 2003, pp. 269-320.
[19] S. S. Haykin, *Neural networks and learning machines*, 3rd ed. Upper Saddle River, NJ: Pearson Education, 2009.
[20] A. Graves, Neural Networks, in Supervised Sequence labelling, with Reccurent Neural Networks, Springer Berlin Heidelberg, 2012, pp. 15-35.
[21] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficul," In Proc. of IEEE Trasactions on Neural Networks, vol. 5, pp. 157-166, 1994 [Online]. Available: IEEE Xplore, https://ieeexplore.ieee.org/document/279181. [Diakses pada 26 Agustus 2020].
[22] R. Pascanu, R. Mikolov, and Y. Bengio, "On the difficulty of training reccurent neural networks," In Proc. of The 30th International Conference on Machine Learning, vol. 28, pp. 1310-1318, 2013 [Online]. Available: https://dl.acm.org/doi/10.5555/3042817.3043083. [Diakses pada 26 Agustus 2020].
[23] M. S. Saputri, R. Mahendra, dan M. Andriani, "Emotional Classification on Indonesian Twitter Dataset," In Proc. of The 2018 International Conference on Asian Language Processing (IALP), 2018, doi: 10.1109/ialp.2018.8629262.
[24] S. Hocreiter and J. Schmidhubler, "Long Short-Term Memory, in Neural Computation," vol. 0, no. 8, pp. 1735-1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
[25] IndoML, Pengenalan Long Short Term Memory (LSTM) dan Gated Reccurent Unit (GRU) – RNN Bagian 2, 2018 [Online]. [Accessed: 31 Agustus 2020].
[26] X. Ma and E. Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF," arXiv:1603.01354v5, 2016 [Online]. Available: https://arxiv.org/abs/1603.01354. [Accessed: 26 Agustus 2020].
[27] C. Dyer, M. Ballasteros, W. Ling, A. Matthews, N. A. Smith, "Transition-Based Dependency Parsing with Stack Long-Short-Term Memory," In Proc. of The 53rd Annual Meeting of the Association for Computational Linguistic and the 7th Joint Conference on Natural Language Processing, Beijing, China, vol. 1, pp. 334-343, 2015 [Online]. Available: https://www.aclweb.org/anthology/P15-1033/. [Accessed: 26 Agustus 2020].
[28] L. Zhang and L. Ma, "Contention based BiLSTM for Answer Selection," In Proc. of The 2017 IEEE International Conference on Information and Automation (ICIA), Macau SAR, China, 2017 [Online]. Available: IEEE Xplore, https://ieeexplore.ieee.org/document/8079049. [Accessed: 26 Agustus 2020].
[29] Y. Kim, "Convolutional Neural Networks for Sentence Classification," arxiv.org/abs/1408.5882, 2014 [Online]. Availabel: https://arxiv.org/abs/1408.5882. [Accessed: 1 September 2020].

Universitas Pertamina
Hak Cipta milik Universitas Pertamina
1. Dilarang mengutip karya tulis, sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.
a. menyebutkan sumber sesuai aidah [cerdekian;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

[30] J. P. C. Chiu and E. Nicholas, "Named Entity Recognition with bidirectional lstm-cnns," arXiv:1511.08308, 2015 [Online]. Available: https://arxiv.org/abs/1511.08308. [Accessed: 26 Agustus 2020].
[31] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, A neural probabilistic language model, Journal of Machine Learning Research (JMLR), vol. 3, pp. 1137-1155.
[32] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vetor space, arXiv:1301.3781, 2013 [Online]. Available: https://arxiv.org/abs/1301.3781. [Accessed: 1 September 2020]
[33] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of word and phrases and their compositionality, arXiv:1310.4546, 2013 [Online]. Available: https://arxiv.org/abs/1310.4546. [Accessed:1 September 2020]
[34] J. Pennington, R. Socher, and C. D. Manning, Glove: Global Vectors for Word Representation, In Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532-1543, Doha, Qatar, 2014 [Online]. Available: https://www.aclweb.org/anthology/D14-1162/. [Accessed: 1 September 2020].
[35] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information Processing and Management, vol. 45, pp. 427-437, 2009.
[36] A. Willyawan, "Named Entity Recognition (NER) Bahasa Indonesia Menggunakan Conditional Random Field dam Pos-Tagging," Skripsi Sarjana, Departemen Teknologi Informasi, Universitas Sumatera Utara, 2018 [Online]. Available: http://repositori.usu.ac.id/handle/123456789/6868. [Accessed: 26 Agustus 2020].
[37] R. Gomez, "Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names,", 2018 [Online]. Available: https://gombru.github.io/2018/05/23/cross_entropy_loss/. [Accessed: 5 September 2020].
[38] C. Perlich, "Learning Curves in Machine Learning", 2011 [Online]. doi: 10.1007/978-0-387-30164-8_452. [Accessed: 5 September 2020].
[39] A. Akhundov, D. Trautmann, and G. Groh, "Sequence Labeling: A Practical Approach", 2018 [Online]. Available: https://arxiv.org/abs/1808.03926. [Accessed: 5 September 2020].
[40] N. Reimers and I. Gurevych, "Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks," 2017 [Online]. Available: https://arxiv.org/pdf/1707.06799.pdf. [Accessed: 5 September 2020].

Universitas Pertamina
Hak Cipta Milik Universitas Pertamina
© Copyright of Universitas Pertamina
1. Dilarang mengutip karya tulis ini, sebagian atau seluruh:
a. menyebutkan sumber sesuai kaidah penulisan;
b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

Universitas
Pertamina

LAMPIRAN

Lampiran 1:

Repositori: https://github.com/gitshrl/chatbot-ner

Hak Cipta Milik Universitas Pertamina Dilindungi Undang-Undang © Copyright of Universitas Pertamina

Universitas Pertamina

1. Dilarang mengutip karya tulis ini, kecuali:
   a. menyebutkan sumber sesuai kaidah kecendekiaan;
   b. pengutipan hanya untuk keperluan pendidikan, penulisan karya ilmiah atau penelitian;
   c. pengutipan tidak merugikan Universitas Pertamina.
2. Dilarang mempublikasikan atau memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa adanya izin dari Universitas Pertamina.

Universitas
Pertamina

Universitas Pertamina	Form TA-2 Bimbingan Tugas Akhir **FAKULTAS SAINS DAN ILMU KOMPUTER** **PROGRAM STUDI ILMU KOMPUTER**

Nama Mahasiswa : Sahrul NIM : 105216044

Nama Pembimbing : Meredita Susanty, M.Sc. NIP : 116020

No. 1	Hari/Tanggal: Senin, 25 November 2019
Hal yang menjadi perhatian: 1. Penentuan batasan dalam penelitian Tugas Akhir 2. Sumber *data set* yang akan digunakan Paraf Pembimbing:	
No. 2	Hari/Tanggal: Selasa, 14 Januari 2020
Hal yang menjadi perhatian: 1. Penggunaan *computing resources* yang akan digunakan pada penelitian dari kampus Paraf Pembimbing:	

Universitas Pertamina	Form TA-2 Bimbingan Tugas Akhir **FAKULTAS SAINS DAN ILMU KOMPUTER** **PROGRAM STUDI ILMU KOMPUTER**

No. 3	Hari/Tanggal: Senin, 3 Maret 2020

Hal yang menjadi perhatian:

1. Akses ke *computing resources* yang akan digunakan pada penelitian dari kampus

Paraf Pembimbing:

No. 4	Hari/Tanggal: Minggu, 30 Mei 2020

Hal yang menjadi perhatian:

1. *Progress review* kemungkinan ikut sidang Juli

Paraf Pembimbing:

Universitas Pertamina	Form TA-2 Bimbingan Tugas Akhir **FAKULTAS SAINS DAN ILMU KOMPUTER** **PROGRAM STUDI ILMU KOMPUTER**

No. 5	Hari/Tanggal: Jumat, 24 Juli 2020

Hal yang menjadi perhatian:

1. Kendala-kendala dalam penelitian yang menghambat pengerjaan laporan
2. Penggunaan *framework* dan pretrained model

Paraf Pembimbing:

No. 6	Hari/Tanggal: Kamis, 6 Agustus 2020

Hal yang menjadi perhatian:

1. Perubahan cakupan dari yang diajukan di proposal

Paraf Pembimbing:

Universitas Pertamina	Form TA-2 Bimbingan Tugas Akhir **FAKULTAS SAINS DAN ILMU KOMPUTER** **PROGRAM STUDI ILMU KOMPUTER**

No. 7	Hari/Tanggal: Sabtu, 8 Agustus 2020
Hal yang menjadi perhatian: 1. Konfrensi yang dapat dikuti terkait dengan topik TA Paraf Pembimbing:	
No. 8	Hari/Tanggal: Senin, 10 Agustus 2020
Hal yang menjadi perhatian: 1. *Review* hasil perubahan cakupan bab 1 Paraf Pembimbing:	

Universitas Pertamina	Form TA-2 Bimbingan Tugas Akhir **FAKULTAS SAINS DAN ILMU KOMPUTER** **PROGRAM STUDI ILMU KOMPUTER**

No. 9	Hari/Tanggal: Selasa, 11 Agustus 2020

Hal yang menjadi perhatian:

1. *Review* hasil perubahan cakupan bab 2

Paraf Pembimbing:

No. 10	Hari/Tanggal: Kamis, 13 Agustus 2020

Hal yang menjadi perhatian:

1. Kendala pada akses *data set*

Paraf Pembimbing: