



SKRIPSI

PEMODELAN *NAMED ENTITY RECOGNITION* UNTUK MENDETEKSI ENTITAS PERISTIWA KEJADIAN PADA BERITA *ONLINE* KABUPATEN SIMEULUE MENGUNAKAN BiLSTM - CNNs

**Disusun Sebagai Syarat Memperoleh Gelar Sarjana Komputer
Prodi Teknik Informatika
Universitas Malikussaleh**

DISUSUN OLEH :

**NAMA : ALFIN GUNAWAN
NIM : 210170021
PRODI : TEKNIK INFORMATIKA**

**JURUSAN INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MALIKUSSALEH
LHOKSEUMAWE
2025**

KATA PENGANTAR

Dengan penuh kerendahan hati, penulis memanjatkan puji syukur ke hadirat Allah SWT. Hanya karena kasih sayang, rahmat, dan hidayah-Nya, penulis diberi kekuatan untuk menyelesaikan skripsi ini, yang berjudul **“Pemodelan *Named Entity Recognition* Untuk Mendeteksi Entitas Peristiwa Kejadian Pada Berita Online Kabupaten Simeulue Menggunakan BiLSTM - CNNs”**.

Setiap lembar yang tertulis dalam skripsi ini adalah hasil dari perjalanan panjang penuh doa, kerja keras, dan dukungan luar biasa dari orang-orang yang hadir di kehidupan penulis. Karena itu, penulis menyadari bahwa karya ini bukanlah hasil dari usaha sendiri, melainkan buah dari cinta, perhatian, dan pengorbanan banyak pihak. Dengan penuh rasa syukur, izinkan penulis mengucapkan terima kasih yang mendalam kepada:

1. Bapak Prof. Dr. Ir. Herman Fithra S.T., M.T., IPM., ASEAN. selaku Rektor Universitas Malikussaleh, yang telah menciptakan lingkungan akademik yang mendukung perkembangan ilmu pengetahuan.
2. Bapak Dr. Muhammad Daud, S.T., M.T, selaku Dekan Fakultas Teknik Universitas Malikussaleh.
3. Bapak Munirul Ula, S.T., M.Eng., Ph.D selaku Ketua Jurusan Informatika Universitas Malikussaleh.
4. Ibu Zara Yunizar, S.Kom., M.Kom selaku Ketua Prodi Teknik Informatika Universitas Malikussaleh.
5. Bapak Rizal, S.Si., M.IT selaku Dosen Pembimbing I, yang dengan ketulusan dan kesabaran membimbing setiap langkah penulis, memberi semangat ketika penulis merasa lelah, serta meluangkan waktu di tengah kesibukannya untuk memastikan penulis berada di jalur yang tepat.
6. Ibu Nunsina, S.T., M.Kom selaku Dosen Pembimbing II, yang tak pernah berhenti memberikan nasihat, arahan, dan dukungan, bahkan di saat penulis merasa ragu pada kemampuan sendiri.
7. Bapak/Ibu Dosen beserta staf karyawan pada Program Studi Teknik Informatika Universitas Malikussaleh, yang telah memberikan inspirasi dan dukungan selama masa perkuliahan.

8. Teristimewa, Ayahanda Abu Marsad dan Ibunda Yulianti, yang menjadi pilar kehidupan penulis. Doa-doa mereka yang tak pernah terputus, kasih sayang yang tak mengenal batas, serta dukungan moril dan materiil mereka adalah alasan utama penulis dapat sampai pada titik ini. Setiap kata dalam proposal ini adalah wujud cinta yang ingin penulis persembahkan untuk mereka
9. Saumi Rahmadani, sosok luar biasa yang selalu hadir di sisi penulis. Terima kasih atas kesabaranmu mendengarkan segala keluh kesah, atas waktu dan tenaga yang kau curahkan, serta atas keyakinanmu yang terus menguatkan langkahku.
10. Kepada keluarga, sahabat dan teman-teman yang tak bisa penulis sebutkan satu per satu. Kehadiran kalian, dukungan, tawa, dan doa yang tulus, telah memberikan penulis kekuatan lebih dari yang bisa diungkapkan dengan kata-kata.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan. Keterbatasan penulis sebagai manusia tentu meninggalkan banyak kekurangan. Oleh karena itu, penulis dengan lapang hati menerima kritik dan saran yang membangun agar dapat terus belajar dan berkembang.

Semoga skripsi ini dapat menjadi langkah kecil yang bermanfaat, tidak hanya untuk penulis, tetapi juga untuk ilmu pengetahuan dan masyarakat. Penulis berharap, apa yang dimulai di sini dapat menjadi awal dari perjalanan yang lebih besar dan bermakna di masa depan.

Lhokseumawe, 25 November 2024

Penulis,

Alfin Gunawan

210170021

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	iv
DAFTAR TABEL	v
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Landasan Teori	5
2.1.1 <i>Named Entity Recognition</i>	5
2.1.2 <i>Web Scraping</i>	6
2.1.5 <i>Convolutional Neural Networks (CNNs)</i>	8
2.1.6 Long Short-Term Memory (LSTM)	8
2.1.7 <i>Bidirectional Long Short-Term Memory (BiLSTM)</i>	9
2.2 Penelitian Terdahulu.....	10
BAB III METODOLOGI PENELITIAN	16
3.1 Waktu dan Tempat Penelitian	16
3.1.1 Waktu Penelitian	16
3.1.2 Tempat Penelitian.....	17
3.2 Sumber Data	17
3.3 Teknik Pengumpulan Data	17
3.4 Skema Kerja Penelitian	18
3.5 Skema Kerja Sistem	21
DAFTAR PUSTAKA	42

DAFTAR GAMBAR

Gambar 2.1 Contoh NER	5
Gambar 2.2 Arsitektur CBOW dan Skip-gram	7
Gambar 2.3 Arsitektur BiLSTM	9
Gambar 3.1 Skema Kerja Penelitian	18
Gambar 3.2 Skema Kerja Sistem	21
Gambar 4.1 Diagram Konteks 1.....	26
Gambar 4.2 DFD Level 1 1.....	27
Gambar 4.3 Form Input URL 1.....	29
Gambar 4.4 Tampilan Hasil Scraping URL 1.....	30
Gambar 4.5 Form Deteksi Entitas 1.....	31
Gambar 4.6 Dataset 1.....	34
Gambar 4.7 Evaluasi Confusion Matrix 1.....	42
Gambar 4.7 Hasil Evaluasi 1.....	43

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	10
Tabel 3.1 Waktu Penelitian	16
Tabel 4.1 Pengujian Dan Deteksi Entitas 1.....	44

BAB I

PENDAHULUAN

1.1 Latar Belakang

Peristiwa merujuk pada kejadian atau rangkaian kejadian bermakna yang terjadi dalam berbagai konteks, seperti sejarah, sastra, atau kehidupan sehari-hari. Secara umum, peristiwa mencakup segala hal yang terjadi, baik biasa maupun luar biasa, yang sering kali berdampak signifikan pada kehidupan masyarakat. Peristiwa-peristiwa luar biasa, seperti bencana alam, konflik sosial, atau perubahan politik, sering menarik perhatian publik karena pengaruhnya yang luas, baik di tingkat lokal maupun internasional. Dalam konteks ini, kebutuhan akan informasi yang cepat dan akurat menjadi semakin penting, terutama bagi masyarakat yang bergantung pada berita *online* untuk mengikuti perkembangan terbaru (Theofany et al., 2024).

Berita online memiliki peran penting dalam menyebarluaskan informasi mengenai peristiwa-peristiwa yang terjadi, baik di tingkat lokal maupun global. Melalui berita *online*, masyarakat dapat memahami situasi terkini, mengambil langkah yang tepat dalam merespons suatu kejadian, dan bahkan membentuk opini publik. Berita online memberikan kecepatan akses informasi dan jangkauan yang lebih luas dibandingkan media cetak atau televisi konvensional, sehingga menjadi sumber utama informasi bagi banyak orang. Berita yang cepat dan akurat berfungsi sebagai sarana edukasi, membantu masyarakat untuk lebih sadar akan kejadian yang mungkin berdampak langsung pada mereka. Meskipun berita *online* berperan penting dalam menyebarkan informasi, tidak semua daerah mendapatkan akses yang memadai terhadap berita ini, terutama di daerah terpencil atau yang kurang mendapat perhatian media arus utama.

Akses informasi di wilayah pelosok seperti Kabupaten Simeulue, Provinsi Aceh, masih terbatas. Kabupaten ini sering kali mengalami peristiwa penting, termasuk bencana alam dan isu sosial, tetapi kurang mendapat perhatian dari media nasional. Masyarakat setempat terpaksa bergantung pada sumber informasi lokal untuk mengetahui perkembangan di daerah mereka, meskipun banyak di antara peristiwa tersebut memiliki dampak yang signifikan. Keterbatasan akses ini

menunjukkan adanya kebutuhan akan metode klasifikasi dan ekstraksi informasi yang mampu menyediakan akses berita yang relevan dengan lebih cepat dan tepat.

Salah satu pendekatan yang dapat mengatasi permasalahan ini adalah *Named Entity Recognition* (NER), sebuah teknik dalam *Text Mining* dan *Natural Language Processing* (NLP) yang bertujuan untuk menemukan dan mengkategorikan entitas dalam teks seperti nama, lokasi, dan organisasi. (Azizi et al., 2023). Melalui penerapan NER, informasi penting dapat lebih mudah diakses dan disaring sehingga berita signifikan di wilayah seperti Simeulue dapat tersebar dengan lebih efisien (Rifani et al., 2019).

Seiring dengan perkembangan teknologi, metode NER terus mengalami peningkatan, terutama dengan dukungan algoritma *machine learning* dan *deep learning*. Beberapa model seperti *Long Short-Term Memory* (LSTM), *Hidden Markov Model* (HMM), dan *Conditional Random Field* (CRF) telah menunjukkan efektivitasnya dalam meningkatkan akurasi ekstraksi entitas (Zahra, 2021). Dalam konteks berita Simeulue, penerapan model *deep learning* yang relevan diharapkan mampu memberikan hasil klasifikasi yang lebih akurat. (Batbaatar & Ryu, 2019).

Teknologi BiLSTM (*Bidirectional Long Short-Term Memory*) dan CNN (*Convolutional Neural Networks*) adalah kombinasi yang efektif untuk deteksi entitas dalam teks. BiLSTM mampu menangkap konteks kalimat secara dua arah, sedangkan CNN dapat mengekstraksi fitur lokal dalam pola kata (Lin & Liu, 2022). Kombinasi ini memungkinkan model mengenali entitas peristiwa dengan lebih akurat. Penelitian (Sukardi et al., 2021) menunjukkan bahwa model BiLSTM-CNNs yang diterapkan pada teks berbahasa Indonesia dengan *embedding word2vec* berhasil mencapai skor F1 sebesar 71.37%, yang mencerminkan efektivitas model ini dalam mendeteksi entitas dari berita formal maupun informal.

Berdasarkan fenomena tersebut, penelitian ini berfokus pada pengembangan model *Named Entity Recognition* (NER) untuk mendeteksi entitas peristiwa dalam berita *online* Kabupaten Simeulue, yang memiliki variasi bahasa formal dan informal. Dengan menerapkan pendekatan *deep learning*, khususnya menggunakan model BiLSTM-CNNs, diharapkan model ini dapat secara efektif menangani variasi bahasa tersebut sehingga informasi yang relevan dapat diakses lebih cepat dan akurat oleh masyarakat Simeulue.

1.2 Rumusan Masalah

Berdasarkan pemaparan latar belakang di atas, masalah utama yang dibahas dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana membangun sebuah sistem berbasis web yang dapat mendeteksi entitas peristiwa dalam berita *online* dengan menggunakan model *Named Entity Recognition* (NER) berbasis BiLSTM dan CNNs?
2. Bagaimana mengukur performa gabungan metode BiLSTM dan CNNs dalam mendeteksi entitas peristiwa kejadian pada suatu berita *online*?

1.3 Batasan Masalah

1. Data yang digunakan hanya artikel berita mengenai peristiwa yang terjadi di Kabupaten Simeulue.
2. Informasi yang diidentifikasi hanya lokasi, peristiwa dan tanggal terjadinya peristiwa.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Membangun sebuah sistem berbasis web yang mampu melakukan *Named Entity Recognition* (NER) pada berita *online* untuk mendeteksi entitas peristiwa, lokasi, dan waktu.
2. Mengembangkan model *Named Entity Recognition* (NER) khusus untuk domain peristiwa kejadian dengan memanfaatkan kombinasi metode BiLSTM dan CNNs.
3. Mengevaluasi kinerja kombinasi BiLSTM dan CNNs dalam mendeteksi entitas peristiwa pada pemberitaan, dengan mengukur *F1-Score* yang dihasilkan

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini meliputi:

1. Memberikan solusi berbasis web yang dapat membantu pengguna dalam mengidentifikasi entitas peristiwa, lokasi, dan waktu dalam berita *online* secara otomatis.

2. Menghasilkan model Named Entity Recognition yang dapat meningkatkan pemahaman terhadap pola entitas dalam berita lokal, khususnya di Kabupaten Simeulue
3. Mendukung pengembangan teknologi *Natural Language Processing* (NLP) untuk ekstraksi informasi dari teks berita, sehingga dapat digunakan untuk berbagai kebutuhan seperti analisis tren berita atau sistem rekomendasi informasi.

1.6 Sistematika Penulisan

Adapun sistematika penulisan pada penelitian ini sebagai berikut:

BAB I PENDAHULUAN

Bagian pendahuluan membahas mengenai latar belakang dilakukannya penelitian ini, serta rumusan masalah, tujuan penelitian, batasan masalah, dan manfaat penelitian.

BAB II LANDASAN TEORI

Bagian ini membahas tentang penelitian sebelumnya dan juga dasar teori yang berkaitan dengan *Named Entity Recognition* (NER).

BAB III METODOLOGI PENELITIAN

Bagian ini menjelaskan tentang langkah – langkah yang dilakukan dalam penelitian ini.

BAB IV HASIL DAN PEMBAHASAN

Bagian ini menjabarkan hasil penelitian mengenai *Named Entity Recognition* (NER), yang mencakup proses implementasi sistem, evaluasi model, serta analisis terhadap kinerja model yang telah dikembangkan. Pembahasan dalam bab ini juga meliputi bagaimana sistem bekerja dalam mendeteksi entitas pada teks berita serta interpretasi hasil yang diperoleh berdasarkan pengujian yang telah dilakukan

BAB V KESIMPULAN

Bab kesimpulan dan saran menjelaskan kesimpulan yang diperoleh dari penelitian ini dan juga memberikan saran agar penelitian selanjutnya dapat dilakukan dengan lebih baik.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 *Named Entity Recognition*

NER (Named Entity Recognition) adalah sub tugas dari *Natural Language Processing* yang mengidentifikasi dan mengkategorikan entitas dengan makna tertentu, seperti nama, tempat, organisasi, waktu, dan angka, ke dalam kategori yang telah ditentukan sebelumnya (Liu et al., 2024). Pendapat lain menwrangkan bahwa NER merupakan tugas dalam komputasi linguistik yang bertujuan untuk mengidentifikasi dan mengklasifikasikan setiap kata dalam dokumen ke dalam salah satu dari berbagai kategori, seperti orang, lokasi, organisasi, tanggal, waktu, persentase, dan nilai moneter. Selain itu, kata-kata tersebut tidak boleh termasuk dalam kategori sebelumnya (Shiraishi et al., 2024). NER juga merupakan tugas penting yang bertujuan untuk mengidentifikasi informasi terstruktur, yang sering kali penuh dengan istilah-istilah teknis yang kompleks dan tingkat variabilitas yang tinggi. Dalam praktiknya, NER menggunakan pendekatan *Text Mining*, yaitu proses untuk mengekstrak pola atau wawasan dari data teks tidak terstruktur, dan *Natural Language Processing* (NLP), cabang kecerdasan buatan yang memungkinkan komputer memahami dan memproses bahasa manusia secara alami (Tjut Adek et al., 2021). Dengan menggabungkan kedua teknik ini, NER dapat memfasilitasi ekstraksi dan analisis informasi penting dengan akurat dan dapat diandalkan. Contoh NER termasuk dalam empat kategori, seperti yang ditunjukkan di bawah ini yaitu, individu (PERSON), organisasi (ORG), tanggal (DATE), dan kelompok agama, kebangsaan, atau politik (NORP).

Gambar 2.1 Contoh Named Entity Recognition)

Sumber (Mcmullen, 2020)

NER dapat digunakan untuk berbagai tujuan, seperti membuat mesin pencari web yang lebih akurat, mengindeks buku secara otomatis, dan memberikan tag entitas nama yang dapat digunakan sebagai langkah preprocessing untuk menyederhanakan pekerjaan seperti penerjemahan mesin. Tugas ekstraksi informasi yang lebih kompleks melibatkan tagger entitas yang dikenal. (Lin & Liu, 2022).

2.1.2 Web Scraping

Web scraping adalah proses mengekstrak data dari web secara terprogram dan mengubahnya menjadi kumpulan data terstruktur. Menurut pendapat lain, *web scraping* adalah metode untuk mendapatkan informasi dari situs web tertentu, yang dapat dilakukan baik secara manual maupun otomatis, dan memungkinkan pengumpulan jumlah data yang lebih besar dalam waktu yang lebih singkat dan dengan cara otomatis yang dapat meminimalkan kerusakan data (Khder, 2021).

Web scraping dapat digunakan untuk berbagai tujuan, seperti mengumpulkan kontak, mengumpulkan ulasan produk, mengumpulkan daftar *real estate*, mengawasi data cuaca, mengidentifikasi perubahan situs web, dan mengintegrasikan data web. (Djiwadikusumah et al., 2021). Program *web scraping* memiliki dua modul penting: modul untuk membuat permintaan HTTP, seperti *Urllib2*, dan modul untuk parsing dan ekstraksi informasi dari kode HTML mentah, seperti *Newspaper*.

2.1.3 Ekstraksi Fitur

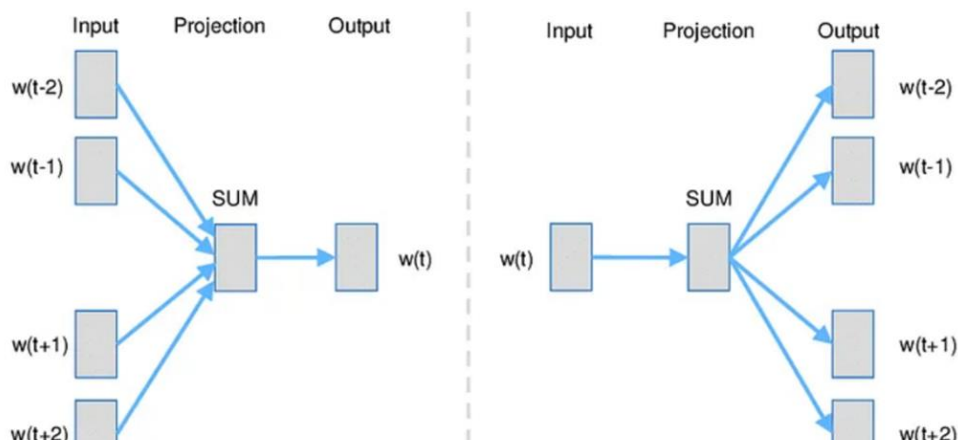
Ekstraksi fitur adalah proses penting dalam NER yang melibatkan pengambilan informasi yang relevan dari data teks untuk diolah lebih lanjut oleh model pembelajaran mesin. Dalam konteks NER, fitur-fitur seperti urutan kata, morfologi, dan konteks semantik dari kata-kata dalam kalimat sangat penting untuk menentukan label entitas dengan akurat. Fitur lokal seperti pola kata, serta fitur global seperti konteks antar-kata dalam kalimat, perlu diidentifikasi dengan baik agar model dapat mengenali entitas secara tepat. Penggunaan teknik *deep learning* seperti BiLSTM dan CNN membantu meningkatkan kemampuan model dalam menangkap fitur lokal dan global dari teks, memberikan hasil yang lebih akurat

dalam deteksi entitas (Sun & Li, 2023). Ada berbagai jenis fitur. Namun, metode Word2Vec digunakan untuk memasukkan kata-kata dalam penelitian ini.

2.1.4 Word2Vec

Word2Vec adalah teknik populer dalam pembelajaran representasi kata yang banyak digunakan dalam tugas NER untuk menghasilkan vektor kata yang merepresentasikan makna semantik dari kata dalam teks. Algoritma ini bekerja dengan mengonversi kata menjadi vektor dengan dimensi tetap, sehingga memungkinkan model untuk memahami konteks dari kata-kata berdasarkan pola kemunculannya dalam kalimat. *Word2Vec* sangat efektif dalam menangkap makna semantik yang tersembunyi di balik kata-kata dan memanfaatkan asosiasi kata untuk membantu model NER dalam mengenali entitas yang lebih kompleks (Liang & Shi, 2023).

Ekstraksi fitur dan penerapan *Word2Vec* dalam NER sangat penting dalam



Gambar 2.2 Arsitektur CBOW dan Skip-gram
Sumber: (Aiensured, 2023)

meningkatkan kemampuan model untuk memahami dan mengenali entitas dalam berbagai jenis teks, terutama dalam bahasa-bahasa yang kaya akan variasi konteks seperti bahasa Indonesia.

2.1.5 *Convolutional Neural Networks (CNNs)*

CNN merupakan salah satu metode *deep learning* yang banyak digunakan dalam berbagai tugas pengenalan pola, termasuk NER. CNN memiliki kemampuan yang unggul dalam menangkap fitur lokal dari data teks, seperti pola kata atau urutan karakter yang berulang, yang sangat berguna dalam mengenali entitas di dalam teks. Dalam konteks NER, CNN berperan dalam melakukan ekstraksi fitur dari teks yang tidak terstruktur dan mengubahnya menjadi representasi yang dapat digunakan untuk klasifikasi entitas (Lv et al., 2023).

Keunggulan CNN dalam NER juga terlihat pada kemampuannya untuk menangkap hubungan spasial dalam data teks. Model CNN mengoptimalkan pengolahan konteks lokal melalui konvolusi, yang penting untuk mengenali entitas yang memiliki batas atau struktur yang sulit dikenali. Contohnya, dalam kasus NER yang bersifat *nested* (entitas tumpang tindih), CNN dapat membantu dalam mendeteksi entitas yang saling beririsan, sebuah fitur yang terbukti meningkatkan kinerja model secara signifikan pada berbagai dataset NER tumpang tindih (Yan et al., 2023).

2.1.6 Long Short-Term Memory (LSTM)

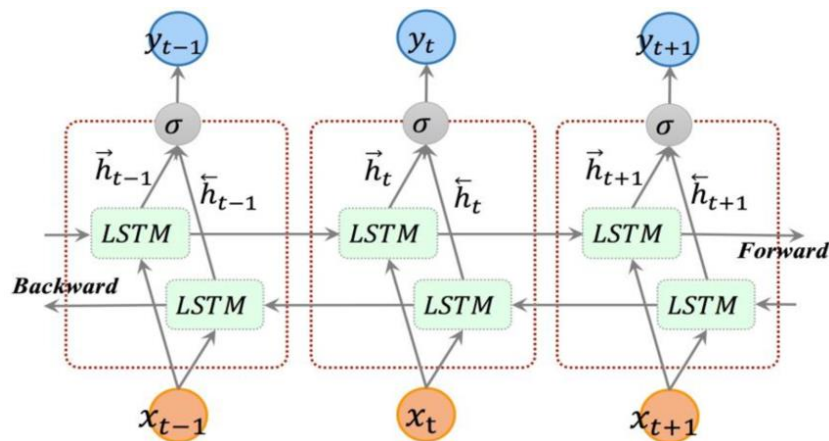
Long Short-Term Memory (LSTM) adalah jenis arsitektur jaringan neural berulang (RNN) yang sangat diakui karena kemampuannya dalam menangani data sekuensial dengan ketergantungan jangka panjang. Dirancang untuk mengatasi masalah gradien meledak atau menghilang yang sering muncul pada RNN konvensional, LSTM memungkinkan pemodelan data sekuensial yang lebih efektif melalui memori sel internal yang memungkinkan informasi bertahan untuk periode waktu yang panjang. Dalam praktiknya, LSTM telah diterapkan dalam berbagai bidang seperti pengenalan suara, pemodelan bahasa, dan prediksi pasar finansial (Lindemann et al., 2021).

Selain itu, pengembangan seperti arsitektur Att-LSTM yang menambahkan mekanisme atensi mampu menangani ketergantungan jangka panjang yang lebih kompleks dalam tugas-tugas klasifikasi sekuensial (Li et al., 2019). Inovasi-inovasi ini menggarisbawahi keunggulan LSTM dalam pemrosesan data sekuensial yang semakin kompleks dan beragam.

2.1.7 Bidirectional Long Short-Term Memory (BiLSTM)

BiLSTM (*Bidirectional Long Short-Term Memory*) adalah varian dari LSTM yang dirancang untuk memproses data dari dua arah, yaitu maju dan mundur, sehingga mampu menangkap konteks sekuensial secara lebih menyeluruh. Dalam penerapan *Named Entity Recognition* (NER), BiLSTM telah menunjukkan keefektifan dalam mengenali entitas dalam teks kompleks dengan memanfaatkan konteks dari kedua arah. Hal ini memungkinkan model untuk mengurangi ambiguitas bahasa dan meningkatkan akurasi dalam pengenalan entitas. (Subowo et al., 2022).

BiLSTM sangat berguna dalam mengatasi tantangan ambigu bahasa dan membantu model untuk secara lebih efektif memahami hubungan antar kata dalam sebuah kalimat. Misalnya, kata-kata yang memiliki arti ganda atau kata-kata yang tergantung pada konteks tertentu dapat lebih mudah diinterpretasi dengan bantuan BiLSTM yang mempertimbangkan informasi dari kedua sisi teks. Hal ini secara signifikan meningkatkan akurasi dalam tugas pengenalan entitas, seperti yang ditunjukkan dalam beberapa studi terkini (Sukardi et al., 2021).



Gambar 2.3 Arsitektur BiLSTM
Sumber: (Subowo et al., 2022)

2.2 Penelitian Terdahulu

Penelitian sebelumnya yang dijadikan referensi untuk penelitian ini dibahas dalam penelitian ini, seperti yang ditunjukkan pada Tabel 2.1 berikut.

Tabel 2.1 Penelitian Terkait

No	Judul Penelitian	Hasil
1	<i>Bidirectional LSTM-CNNs</i> Untuk Ekstraksi <i>Entity</i> Lokasi Kebakaran Pada Berita Online Berbahasa Indonesia (Putra & Kurniawan, 2021)	Penelitian ini menunjukkan model Ner dengan BLSTM-CNNs memiliki performa yang baik k berdasarkan hasil perhitungan F1-score, presisi dan recall. Kemudian, dilakukan pemetaan berdasarkan entity lokasi yang terdapat dalam artikel berita online hasil klasifikasi menggunakan model NER dengan BLSTM-CNNs.
2	Pemodelan <i>Named Entity Recognition</i> Pada Artikel Wisata Dengan Metode <i>Bidirectional Long Short-Term Memory Dan Conditional Random Fields</i> (Zahra, 2021)	Penelitian ini menunjukkan bahwa model yang dibuat dapat mendeteksi berbagai entitas tempat wisata, tetapi masih ada banyak kesalahan. F1-Score tertinggi sebesar 75,25% dihasilkan dari berbagai skenario model yang diuji.
3	<i>Named entity recognition for extracting concept in ontology building on Indonesian language using end-to-end bidirectional long short term memory</i> (Santoso et al., 2021).	Penelitian ini menunjukkan bahwa model NER berkerja dengan baik sehingga dapat mengerkstrak entitas yang ditargetkan, seperti orang, lokasi, organisasi dan lain-lain. Model ini mencapai F1-Score terbaik untuk <i>Part-of-Speech Tagging</i> sebesar 91,79% dan <i>Named Entity Recognition</i> sebesar 83,18%.

Tabel 2.1 Penelitian Terkait (Lanjutan)

No	Judul Penelitian	Hasil
4	Implementasi Metode <i>Textrank</i> Dan <i>Named Entity Recognition</i> Untuk Ekstraksi Kata Kunci Pada Media <i>Online</i> Berita (Theofany et al., 2024).	Dari penelitian ini dapat dibuktikan, kinerja gabungan metode <i>TextRank</i> dan NER dalam mengekstraksi kata kunci dari artikel berita lebih baik dibandingkan dengan penggunaan <i>TextRank</i> secara tunggal. Hal ini dapat dilihat dari nilai rata-rata <i>recall</i> , <i>precision</i> , <i>f-measure</i> , dan <i>accuracy</i> yang dihasilkan dari eksperimen dengan 300 artikel dan <i>weight multiplier</i> 2 dengan nilai masing-masing 0.652, 0.645, 0.648, 0.505. Secara kesimpulan, integrasi <i>TextRank</i> dan NER dapat secara signifikan meningkatkan kualitas ekstraksi kata kunci dari artikel berita <i>online</i> .
5	<i>Named Entity Recognition</i> Menggunakan Metode <i>Bidirectional Lstm-Crf</i> Pada Teks Bahasa Indonesia (Permana et al., 2022)	Berdasarkan hasil pengujian menggunakan data training sebanyak 25.709 kata dan testing 9.406, metode <i>bidirectional LSTM-CRF</i> memperoleh akurasi sebesar 87.77%.
6	Ekstraksi Informasi pada Data <i>Logbook</i> KKN Mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya Malang menggunakan Metode NER (<i>Named Entity Recognition</i>) (Azizi et al., 2023).	Penelitian ini menghasilkan informasi yang sudah didapat dari ekstraksi informasi dan melakukan peningkatan keakurasi NER pada <i>library polyglot</i> dalam melakukan ekstraksi informasi <i>logbook</i> data KKN.

Tabel 2.1 Penelitian Terkait (Lanjutan)

No	Judul Penelitian	Hasil
7	<i>Improving Indonesian Named Entity Recognition for Domain Zakat Using Conditional Random Fields</i> (Widiyanti et al., 2023).	Hasil penelitian menunjukkan bahwa informasi ini dapat meningkatkan efisiensi dan transparansi sistem Zakat. Meningkatkan efisiensi dan transparansi sistem Zakat serta mendukung penelitian dan analisis tentang topik-topik terkait Zakat. Evaluasi kinerja rata-rata dari model Indonesia-NER menunjukkan presisi sebesar 0,902, <i>recall</i> sebesar 0,834, dan nilai F1-score sebesar 0,867.
8	<i>A Hybrid Cnn-Bilstm Model For Drug Named Entity Recognition</i> (Fudholi et al., 2022)	Hasil penelitian ini menunjukkan bahwa pada percobaan yang dilakukan, pencapaian terbaik diperoleh oleh salah satu model dalam hal skor F1-nya. Model tersebut menggunakan 2 lapisan CNN dengan ukuran kernel 7, filter CNN 50, satu lapisan LSTM dengan 200 unit tersembunyi, dan tambahan fitur berbasis <i>chunk tag</i> . Model ini mencapai skor f1 sebesar 0,892, presisi 0.881, dan <i>recall</i> 0.903.
9	<i>Named Entity Recognition of Tourist Destinations Reviews in the Indonesian Language</i> (Hidayatullah et al., 2023)	Dalam penelitian ini, diterapkan beberapa skenario <i>hyperparameter</i> menggunakan BLSTM. Berdasarkan hasil percobaan, mendapatkan nilai F1 rata-rata terbaik sebesar 94.3%.

Tabel 2.1 Penelitian Terkait (Lanjutan)

No	Judul Penelitian	Hasil
10	<i>Indonesian disaster named entity recognition from multi source information using bidirectional LSTM (BiLSTM)</i> (Shidik et al., 2024)	Hasil penelitian menunjukkan peningkatan performa BiLSTM menggunakan optimasi Adam dan korpus yang seimbang. Indikator kinerja yang dicapai adalah 93,4%, 82,4%, dan 87,5% untuk presisi, <i>recall</i> , dan F1-score. Jaringan BiLSTM menangkap ketergantungan jarak jauh dalam data berurutan yang disediakan oleh NER. <i>Oversampling</i> memastikan bahwa model NER yang diusulkan dapat secara tepat mengenali semua entitas dan mengurangi hasil yang bias. Dengan demikian, metode BiLSTM dapat mengidentifikasi entitas dalam korpus tekstual berita daring terkait bencana di Indonesia dengan lebih baik. Berita <i>online</i> terkait bencana di Indonesia.

Penelitian sebelumnya menunjukkan keberhasilan model deep learning seperti BiLSTM, CNN, dan CRF dalam tugas Named Entity Recognition (NER) pada teks berbahasa Indonesia, yang relevan dengan penelitian ini. Penelitian (Putra & Kurniawan, 2021) memiliki kesamaan dengan penelitian ini, yaitu sama-sama menggunakan kombinasi BiLSTM dan CNN untuk NER pada berita online. Kedua penelitian memanfaatkan potensi arsitektur ini untuk mengidentifikasi entitas dalam teks, namun fokusnya berbeda: penelitian mereka berfokus pada entitas lokasi kebakaran, sedangkan penelitian ini mencakup entitas peristiwa yang lebih luas, seperti bencana alam, kecelakaan, atau peristiwa sosial.

Penelitian (Zahra, 2021) dan penelitian ini memiliki kesamaan dalam penggunaan model deep learning, khususnya arsitektur Bidirectional Long Short-Term Memory (BiLSTM), untuk mencapai tujuan NER. Perbedaannya terletak

pada sumber data, di mana Zahra menggunakan artikel wisata internasional berbahasa Inggris, sedangkan penelitian ini menggunakan berita online lokal berbahasa Indonesia, yang menghadirkan tantangan tersendiri dalam membangun model.

Penelitian (Santoso et al., 2021) dan penelitian ini memiliki kesamaan dalam penggunaan model deep learning dengan arsitektur Bidirectional Long Short-Term Memory (BiLSTM). Namun, perbedaannya terletak pada pendekatan, penelitian Santoso menggunakan BiLSTM secara end-to-end untuk ekstraksi konsep, sedangkan penelitian ini mengombinasikan BiLSTM dengan CNN, di mana CNN berperan dalam menangkap fitur lokal dalam teks.

Penelitian (Theofany et al., 2024) dan penelitian ini sama-sama berfokus pada pemrosesan bahasa alami pada teks berita online. Namun, perbedaannya terletak pada fokus utama, penelitian Theofany berfokus pada ekstraksi kata kunci sebagai representasi teks, sementara penelitian ini berfokus pada deteksi entitas bernama, khususnya entitas peristiwa kejadian.

Penelitian (Permana et al., 2022) dan penelitian ini sama-sama menggunakan model *deep learning* untuk mencapai tujuannya. Perbedaannya terletak pada model tambahan yang digunakan, di mana penelitian Permana menggunakan *Conditional Random Fields* (CRF) sebagai lapisan output, sementara penelitian ini menggunakan *Convolutional Neural Networks* (CNNs).

Penelitian (Azizi et al., 2023) dan penelitian ini sama-sama memanfaatkan Named Entity Recognition (NER) untuk mengekstrak informasi spesifik dari teks. Perbedaannya terletak pada sumber data, penelitian Azizi menggunakan data logbook KKN mahasiswa, sementara penelitian ini menggunakan berita online, yang memiliki struktur teks lebih jelas dan terorganisir.

Penelitian (Widiyanti et al., 2023) dan penelitian ini sama-sama menggunakan metode *Named Entity Recognition* (NER) sebagai alat utama. Perbedaannya terletak pada domain teks, penelitian Widiyanti berfokus pada domain zakat dengan terminologi spesifik, sementara penelitian ini berfokus pada berita online di Kabupaten Simeulue yang memiliki karakteristik bahasa dan peristiwa berbeda.

Penelitian (Fudholi et al., 2022) dan penelitian ini sama-sama melibatkan *Convolutional Neural Networks* (CNNs) untuk menangkap fitur lokal dalam teks. Perbedaannya terletak pada domain teks yaitu penelitian Fudholi berfokus pada domain medis, khususnya teks terkait obat-obatan, sementara penelitian ini berfokus pada berita online di Kabupaten Simeulue.

Penelitian (Hidayatullah et al., 2023) dan penelitian ini sama-sama menggunakan arsitektur *Recurrent Neural Network* (RNN), yaitu *Bidirectional Long Short-Term Memory* (BiLSTM). Perbedaannya terletak pada jenis entitas yang diteliti: penelitian Hidayatullah berfokus pada entitas tempat wisata, seperti nama hotel, restoran, objek wisata, dan lokasi geografis, sementara penelitian ini berfokus pada entitas peristiwa kejadian, seperti bencana alam, kecelakaan, atau peristiwa sosial.

Penelitian (Shidik et al., 2024) dan penelitian ini sama-sama menggunakan model *deep learning* BiLSTM sebagai fondasi untuk melakukan NER. Perbedaannya terletak pada model tambahan: penelitian Shidik hanya menggunakan arsitektur BiLSTM, sementara penelitian ini menggabungkan model tambahan *Convolutional Neural Networks* (CNNs).

Penelitian terdahulu memberikan landasan kuat untuk penggunaan BiLSTM-CNNs dalam penelitian ini. Penelitian ini berbeda dengan penelitian sebelumnya meskipun menggunakan model *deep learning* seperti BiLSTM-CNNs untuk tugas *Named Entity Recognition* (NER). Fokus penelitian ini adalah pada deteksi entitas peristiwa kejadian, sementara penelitian lain lebih fokus pada entitas lokasi atau tempat wisata. Pendekatan ini diharapkan lebih efektif untuk menangani tantangan dalam pemrosesan teks berita lokal

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

3.1.1 Waktu Penelitian

Penelitian memerlukan waktu dan jadwal yang terstruktur, mencakup proses bimbingan, seminar proposal, seminar hasil, hingga sidang skripsi. Gambaran waktu dan jadwal penelitian dapat dilihat pada tabel 3.1 berikut:

Tabel 3.1 Waktu Penelitian

Kegiatan	2024 – 2025								
	Okt 2024	Nov 2024	Des 2024	Jan 2025	Feb 2025	Mar 2025	Apr 2025	Mei 2025	Jun 2025
Tahap 1									
Pengajuan Judul									
Penyusunan Proposal									
Konsultasi									
Seminar Proposal									
Tahap 2									
Pengumpulan Data									
Pengolahan Data									
Pengusunan Laporan TA									
Seminar Hasil									
Tahap 3									
Sidang TA									
Revisi TA									

3.1.2 Tempat Penelitian

Penelitian ini dilaksanakan di Laboratorium Teknik Informatika atau tempat lain yang mendukung penggunaan perangkat lunak yang diperlukan. Laboratorium dipilih untuk memastikan lingkungan penelitian yang kondusif, terkontrol, dan memiliki sumber daya teknologi yang memadai untuk mendukung seluruh tahap penelitian, mulai dari pengumpulan data hingga evaluasi model. Selain itu, lokasi ini juga memberikan fleksibilitas bagi peneliti untuk melakukan pengujian dan pemrosesan data dengan optimal. Jika laboratorium tidak tersedia, penelitian juga dapat dilakukan di lokasi alternatif seperti ruang kerja pribadi, *co-working space*, atau tempat lain yang memiliki fasilitas serupa. Namun, pemilihan lokasi tetap memastikan adanya dukungan teknologi yang memadai untuk kelancaran proses penelitian.

3.2 Sumber Data

Data penelitian biasanya terbagi menjadi dua kategori utama, yaitu data primer dan data sekunder. Pada penelitian ini, penulis menggunakan data sekunder yang diperoleh dari situs-situs berita online seperti kabarsimeulue.id, simeuluekab.go.id, serambinews.com, kompas.com, liputan6.com, suara.com, tempo.co, dan detik.com. Data ini berupa artikel berita yang berisi informasi tentang peristiwa yang terjadi di Kabupaten Simeulue. Pemilihan sumber data ini didasarkan pada relevansi dan cakupan berita yang mencakup berbagai peristiwa penting di Kabupaten Simeulue. Artikel-artikel berita tersebut akan dijadikan dasar untuk membangun model *Named Entity Recognition* (NER) yang difokuskan pada deteksi entitas

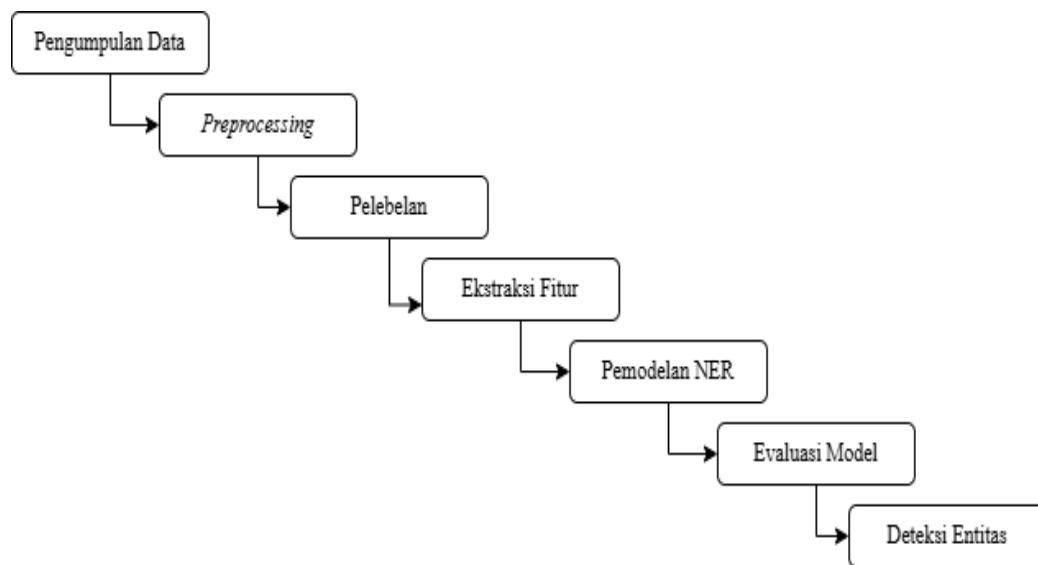
3.3 Teknik Pengumpulan Data

Teknik pengumpulan data dalam penelitian ini dilakukan melalui metode *web scraping* menggunakan modul *Newspaper* pada *Python*. Metode ini bertujuan untuk mengotomatisasi proses pengambilan artikel-artikel berita dari situs-situs yang telah ditentukan sebagai sumber data. Data hasil *scraping* kemudian disimpan dalam format *.txt*, yang selanjutnya akan melalui tahap *preprocessing* dan pelabelan. Hasil akhirnya digunakan sebagai *dataset* untuk proses pelatihan dan

pengujian model *Named Entity Recognition* (NER) yang dikembangkan dalam penelitian ini.

3.4 Skema Kerja Penelitian

Dalam penelitian ini sudah di susun berbagai langkah – langkah penelitian yang nantinya akan dilakukan secara sistematis. Pada gambar 3.1 ini merupakan urutan dari penelitian yang akan dilakukan oleh penulis.



Gambar 3.1 Skema Kerja Penelitian

Tahapan penelitian menggambarkan proses penelitian yang akan dijalankan dan menggambarkan penelitian secara keseluruhan. Tahap-tahap penelitian yaitu:

1. Pengumpulan data

Pada tahap ini peneliti mengumpulkan data – data yang akan digunakan dari berita *online* mengenai peristiwa yang terjadi di Kabupaten Simeulue.

2. *Preprocessing*

Pada penelitian ini, data yang telah berhasil diekstrak akan melalui proses *preprocessing* untuk mempersiapkan teks masukan agar lebih mudah dipahami oleh komputer. Proses *preprocessing* meliputi beberapa tahap, yaitu:

a. Penghapusan URL

URL dalam teks berita dihapus karena tidak memberikan kontribusi signifikan terhadap analisis. Informasi ini dianggap tidak relevan dengan proses pemodelan *Named Entity Recognition* (NER).

b. *Case Folding*

Case folding bertujuan untuk menstandarkan teks dengan menghapus tanda baca dan karakter yang tidak diperlukan dalam analisis. Pada tahap ini, karakter yang diperbolehkan hanyalah tanda titik (“.”), koma (“,”), garis miring (“/”), tanda kurung (“()”), huruf kecil dari “a” hingga “z”, dan angka dari “0” hingga “9”. Selain itu, karakter kosong atau *whitespace* juga dihilangkan untuk menjaga konsistensi teks.

c. Tokenisasi

Tokenisasi adalah proses membagi dokumen teks menjadi bagian-bagian yang lebih kecil, seperti kalimat atau kata, yang disebut token. Proses ini bertujuan untuk mengonversi teks menjadi unit-unit analisis yang lebih sederhana, sehingga model dapat memahami dan menganalisis setiap bagian teks secara lebih efektif.

3. Pelebelan data

Proses pelebelan dilakukan menggunakan skema *NER Tagging*. Setiap token diberi label berdasarkan perannya dalam entitas. Contohnya, token yang entitasnya lokasi diberi label *Loc*, token yang memiliki peran suatu kejadian atau peristiwa diberi label *Event*, token yang menunjukkan waktu dan tanggal diberi label *Time* dan token di luar entitas diberi label *O*. Skema ini membantu model dalam mengenali struktur entitas dengan lebih baik

4. Ekstraksi fitur

Setelah data diproses, langkah selanjutnya adalah mengekstraksi fitur menggunakan metode *Word2Vec*. Metode ini mengubah kata-kata menjadi representasi numerik dalam bentuk vektor berdimensi. Vektor ini memetakan hubungan antar kata berdasarkan konteksnya dalam teks. Dengan representasi ini, model dapat mengenali hubungan semantik antara kata-kata yang berbeda, seperti lokasi atau waktu peristiwa.

5. Pemodelan *Named Entity Recognition*

Sebelum membangun model, data dibagi menjadi *data latih*, *data validasi*, dan *data uji*. Data latih digunakan untuk melatih model, sementara data validasi berfungsi untuk memvalidasi model guna mencegah *overfitting*. *Overfitting* adalah kondisi di mana model terlalu baik dalam mempelajari data latih namun memiliki performa rendah pada data baru. Data yang tersisa digunakan sebagai data uji untuk mengevaluasi performa model setelah proses pelatihan.

Selanjutnya, model dibangun menggunakan kombinasi BiLSTM dan CNNs. Teks diubah menjadi vektor menggunakan Word2Vec, yang kemudian dilatih untuk membentuk kamus kata. Kamus tersebut akan digunakan sebagai bobot pada lapisan embedding untuk memetakan setiap kata ke dalam vektor sesuai representasinya. Setelah melalui lapisan embedding, vektor masuk ke lapisan BiLSTM, yang terdiri dari forward dan backward LSTM untuk menangkap konteks masa lalu dan masa depan dalam kalimat. Keluaran dari lapisan BiLSTM berupa matriks representasi konteks, yang kemudian diproyeksikan melalui lapisan CNNs untuk mengekstraksi fitur-fitur lokal, sehingga menghasilkan skor prediksi untuk setiap label.

6. Evaluasi model

Pada tahap ini akan dilakukan evaluasi model yang telah dilatih. Sebagai parameter pengukuran kinerja model, penelitian ini akan menggunakan *F1-Score*, yang diperoleh dari rata-rata harmonik precision dan recall, seperti yang ditunjukkan pada Persamaan 3.1.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \dots \dots \dots (3.1)$$

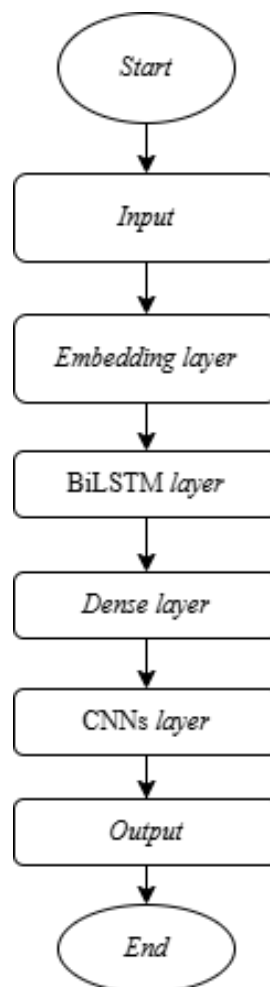
- a. *Precision* (presisi) mengukur seberapa banyak prediksi yang benar-benar relevan di antara seluruh prediksi positif. Nilai precision yang tinggi menunjukkan bahwa model tidak sering membuat kesalahan dalam mengidentifikasi entitas yang tidak relevan.

- b. *Recall* (recall) mengukur seberapa banyak dari keseluruhan entitas yang relevan telah berhasil diidentifikasi oleh model. Nilai recall yang tinggi berarti model mampu menemukan sebagian besar entitas penting dalam teks.

7. Deteksi entitas

Tahap akhir adalah implementasi model untuk mendeteksi entitas seperti lokasi, waktu, dan jenis peristiwa dalam teks berita. Model menggunakan hasil evaluasi terbaik untuk mendeteksi entitas dengan tingkat akurasi tinggi. Proses ini memungkinkan informasi penting dari berita diidentifikasi dengan cepat, yang kemudian dapat dimanfaatkan untuk berbagai kebutuhan analisis atau pelaporan.

3.5 Skema Kerja Sistem



Gambar 3.2 Skema Kerja Sistem

Skema sistem ini menggambarkan alur kerja dalam proses analisis teks untuk mendeteksi entitas yang terkandung di dalamnya, seperti lokasi, peristiwa, dan tanggal. Setiap tahapan dalam sistem ini bekerja secara bertahap, mengolah teks dari input hingga menghasilkan output yang dapat mengidentifikasi dan memberi label pada entitas secara akurat. Adapun tahapan-tahapan tersebut adalah sebagai berikut:

1. Input

Proses dimulai dengan memasukkan data berupa teks, seperti kalimat atau kumpulan kata, yang akan dianalisis untuk mengidentifikasi entitas, misalnya entitas "Loc," "Event," atau "Time".

2. Embedding Layer

Teks yang masuk akan diubah menjadi representasi vektor melalui *Embedding Layer* menggunakan metode seperti Word2Vec. Representasi ini memungkinkan sistem memahami hubungan antar kata dalam ruang berdimensi tertentu.

3. BiLSTM Layer

Representasi vektor tersebut diproses oleh *BiLSTM Layer*, yang menganalisis teks secara dua arah:

- a. Dari awal hingga akhir kalimat (forward).
- b. Dari akhir kembali ke awal (backward). Pendekatan ini memungkinkan model menangkap konteks kata berdasarkan informasi dari masa lalu dan masa depan dalam satu kalimat.

4. Dense Layer

Hasil dari *BiLSTM Layer* diteruskan ke *Dense Layer*, yang bertugas memproyeksikan hasil tersebut ke dalam dimensi yang sesuai dengan jumlah kelas atau label yang akan diprediksi.

5. CNN Layer

Data yang telah diproses di *Dense Layer* diteruskan ke *CNN Layer*, yang berfungsi untuk mengekstraksi pola-pola lokal dari teks, seperti urutan kata atau n-gram yang sering muncul dan relevan dalam proses pelabelan.

6. Output

Sistem menghasilkan output berupa label atau kategori untuk setiap kata dalam teks input. Misalnya, dalam kalimat "Kebakaran terjadi di Simeulue pada 21 Mei 2023":

- a. "Simeulue" diberi label sebagai *Loc*,
- b. "21 Mei 2023" sebagai *Time*,
- c. "Kebakaran" sebagai *Event*.

Dengan skema ini, sistem dapat mengenali dan melabeli entitas dalam teks secara akurat, mendukung pengembangan model *Named Entity Recognition* (NER) yang andal.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Hasil dari penelitian ini yakni sebuah sistem yang dapat deteksi entitas suatu peristiwa dalam berita *online* Kabupaten Simeulue dengan menerapkan metode *Bidirectional LSTM* (BiLSTM) dan *Convolutional Neural Networks* (CNNs). Sistem ini berfungsi dengan menerima input berupa URL dari berita yang ingin deteksi, lalu mengklasifikasikan entitas pada berita tersebut kedalam kategori *event*, *loc* dan *time*. Model yang dikembangkan menggunakan metode BiLSTM dan CNNs menunjukkan kinerja yang baik dengan nilai *F1-Score* yang memadai.

4.2 Analisa Sistem

Sistem yang diimplementasikan dalam penelitian ini adalah sebuah sistem yang secara otomatis mendeteksi entitas *event*, *location* (loc), dan *time* dalam berita *online* berdasarkan *input* URL dari berita yang ingin diperiksa. Program ini dibangun berbasis web menggunakan bahasa *Python*. Tahapan dalam membangun sistem ini dimulai dari pengumpulan data, yang diperoleh melalui web *scraping* dari beberapa situs berita *online* di Kabupaten Simeulue. *Dataset* ini berisi berita yang telah dilabeli dengan entitas *event*, *location*, dan *time*, yang menjadi dasar untuk deteksi dalam sistem ini.

Proses dimulai dengan *preprocessing* data, yaitu membersihkan teks dari karakter yang tidak relevan serta melakukan *tokenisasi* untuk mempersiapkan data bagi model. Data kemudian dibagi menjadi data latih dan data uji, di mana data latih digunakan untuk melatih model agar mampu mengenali pola-pola dalam teks berita, sedangkan data uji digunakan untuk mengukur performa model dalam mendeteksi entitas secara akurat.

Untuk ekstraksi fitur, sistem menggunakan *word embedding* berbasis *Word2Vec* atau *FastText* guna mengubah teks menjadi representasi numerik yang dapat dipahami oleh model. Model yang digunakan untuk klasifikasi adalah

BiLSTM dan CNNs, yang menggabungkan kekuatan *Bidirectional Long Short-Term Memory* (BiLSTM) untuk menangkap konteks dalam teks serta *Convolutional Neural Networks* (CNNs) untuk mengenali pola entitas. Setelah model dilatih, berita baru dapat diproses dengan memasukkan URL berita, lalu sistem akan mengambil teks berita, mengekstrak informasi, dan mengidentifikasi entitas event, location, serta time dalam teks.

Evaluasi kinerja dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score untuk memastikan keandalan model dalam mendeteksi entitas yang relevan dengan peristiwa dalam berita *online*.

4.3 Analisa Data

Untuk mendeteksi entitas dalam berita *online*, *dataset* yang digunakan berbentuk teks yang diperoleh melalui web *scraping* dari beberapa situs berita Kabupaten Simeulue, seperti *detik.com*, *kompas.com*, *liputan6.com*, *sumut.suara.com*, *tvonenews.com*, *tribratanewssimeulue.com*, *popularitas.com*, *aceh.inews.id*, *aceh.tribunnews.com*, dan *m.jpnn.com*. *Dataset* ini berisi teks berita yang telah dilabel dengan tiga jenis entitas, yaitu *event*, *location*, dan *time*.

Data yang digunakan dalam penelitian ini berjumlah 20.782 token, yang terdiri dari 941 kalimat hasil ekstraksi dari 100 berita. Setiap berita telah melalui proses pelebelan manual untuk menandai keberadaan entitas yang relevan. *Dataset* ini kemudian dibagi menjadi data latih dan data uji dengan proporsi 80% atau 752 kalimat sebagai data latih dan 20% atau 189 kalimat sebagai data uji. Pembagian ini bertujuan untuk melatih model NER agar mampu mengenali pola dalam teks dan mengevaluasi kinerjanya menggunakan data uji.

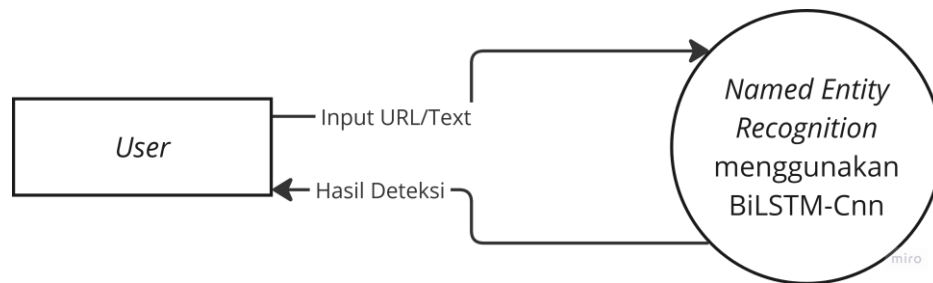
Selain itu, data uji juga mencakup berita terbaru yang diambil dari URL situs berita yang belum pernah digunakan dalam pelatihan model. URL berita ini kemudian diproses untuk diambil teksnya dan dianalisis menggunakan model NER yang telah dilatih. Sistem secara otomatis akan mengekstrak dan mengidentifikasi entitas *event*, *location*, dan *time* dari teks berita, memungkinkan analisis lebih lanjut terhadap informasi yang terkandung dalam berita *online* Kabupaten Simeulue.

4.4 Perancangan Sistem

Perancangan sistem adalah sebuah kegiatan merancang dan menentukan cara mengolah sistem dari hasil analisa sistem. Tujuan dari perancangan sistem yaitu untuk memenuhi kebutuhan pengguna sistem dan untuk memberikan gambaran yang jelas berdasarkan rancang yang ingin dibangun.

4.4.1 Diagram Konteks

Pada diagram konteks ini menjelaskan gambaran umum pada sistem yang dibangun.



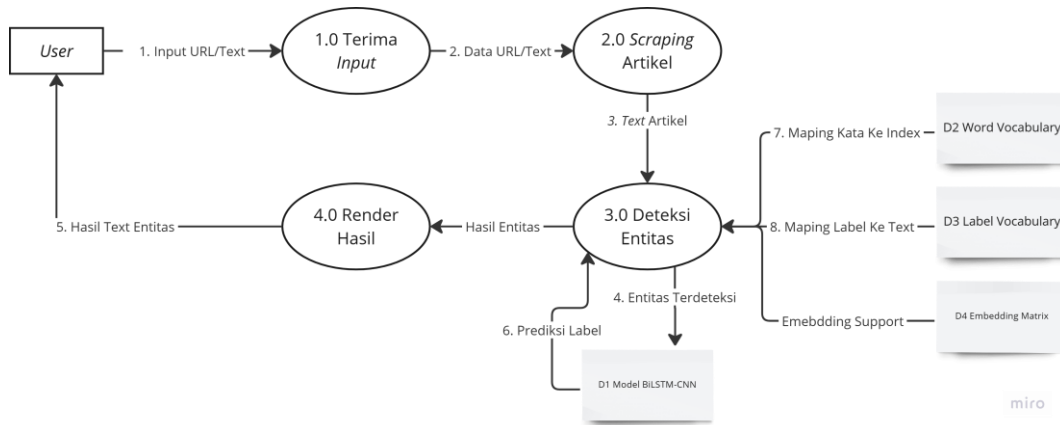
Gambar 4.1 Diagram Konteks 1

Keterangan:

Diagram konteks di atas menggambarkan interaksi antara pengguna dan sistem deteksi entitas berbasis model BiLSTM-CNN untuk *Named Entity Recognition* (NER). Pada diagram ini, Pengguna bertindak sebagai entitas eksternal yang memberikan input berupa URL atau teks ke dalam sistem. Sistem Aplikasi NER berfungsi untuk memproses *input* tersebut dengan menggunakan model NER BiLSTM-CNN, yang bertugas mendeteksi entitas dalam teks. Setelah sistem memproses data, hasil yang dikembalikan berupa teks yang telah dianotasi dengan entitas atau daftar entitas yang terdeteksi, yang kemudian disampaikan kembali kepada pengguna. Aliran data antara pengguna dan sistem berlangsung melalui dua tahap utama, pertama, pengguna memberikan *input* ke sistem, dan kedua, sistem mengembalikan hasil deteksi entitas ke pengguna. Diagram ini memberikan gambaran umum tentang bagaimana sistem bekerja dan batasan antara interaksi pengguna dengan aplikasi deteksi entitas.

4.4.2 DFD Level 1 Detail Proses

Pada DFD Level 1 ini menjelaskan detail proses alur sistem yang dibangun.



Gambar 4.2 DFD Level 1 2

Keterangan:

Pada sistem deteksi entitas ini, proses diawali dengan Proses 1.0: *Terima Input*, di mana pengguna mengirimkan *input* berupa URL atau teks melalui *form* HTML. Aliran data dalam tahap ini mengalir dari pengguna ke sistem sebagai *Input* URL/Teks yang akan diproses lebih lanjut.

Jika *input* yang diberikan berupa URL, maka sistem akan masuk ke Proses 2.0: *Scraping Artikel* menggunakan *library newspaper*. Proses ini bertugas mengambil konten artikel dari URL yang diberikan. Setelah *scraping* selesai, *output* berupa Teks Artikel dikirim ke Proses 3.0: *Deteksi Entitas*.

Pada Proses 3.0: *Deteksi Entitas*, sistem akan memproses teks untuk mengidentifikasi entitas di dalamnya. Proses ini mencakup beberapa tahapan, seperti normalisasi teks, tokenisasi teks, konversi token ke indeks menggunakan D2: *Word Vocabulary*, serta *padding token* agar sesuai dengan format yang diterima oleh model. Setelah itu, sistem melakukan prediksi entitas menggunakan D1: *Model BiLSTM-CNN*, kemudian hasil prediksi dikonversi kembali menjadi teks menggunakan D3: *Label Vocabulary*. Aliran data dalam tahap ini melibatkan Teks Artikel dari Proses 2.0, Prediksi Label dari D1, *Mapping* Kata ke Indeks dari D2,

dan *Mapping Label* ke Teks dari D3. *Output* dari proses ini berupa Entitas Terdeteksi, yang akan diteruskan ke tahap selanjutnya.

Pada Proses 4.0: *Render Hasil*, hasil yang telah diproses, baik dalam bentuk teks artikel atau entitas yang terdeteksi, akan ditampilkan kepada pengguna melalui tampilan berbasis *template* HTML. Aliran data pada tahap ini mengalir dari Proses 4.0 ke Pengguna sebagai hasil akhir yang dapat dilihat langsung oleh pengguna.

Selain proses utama, sistem juga bergantung pada beberapa data *store* (D1-D4) yang digunakan dalam proses deteksi entitas. D1: Model BiLSTM-CNN menyimpan model *machine learning* yang digunakan untuk prediksi entitas. D2: *Word Vocabulary* berisi daftar kata yang digunakan untuk mengonversi teks menjadi indeks numerik. D3: *Label Vocabulary* menyimpan informasi terkait label entitas untuk mengubah hasil prediksi ke dalam bentuk teks yang bisa dibaca manusia. D4: *Embedding Matrix* digunakan untuk menyimpan representasi vektor kata agar model dapat memahami hubungan antar kata dalam teks.

Dengan aliran data yang terstruktur ini, sistem dapat memproses *input* dari pengguna, melakukan deteksi entitas dengan model *machine learning*, dan menampilkan hasilnya dalam antarmuka yang mudah dipahami.

4.5 Implementasi Sistem

Tahapan implementasi sistem terdiri dari serangkaian langkah yang diperlukan dalam pengembangan perangkat lunak yang telah dirancang, dianalisis, dan dikembangkan. Setelah seluruh proses pengembangan selesai, sistem akan melewati tahap pengujian guna memastikan bahwa sistem tersebut berfungsi sesuai dengan spesifikasi serta memenuhi standar kelayakan yang telah ditetapkan.

Implementasi merupakan tahap krusial di mana sistem mulai diterapkan dalam lingkungan sebenarnya. Pada tahap ini, dilakukan penyesuaian terhadap lingkungan operasional, infrastruktur yang dibutuhkan, serta prosedur penggunaan sistem. Hal ini bertujuan agar sistem dapat berjalan secara optimal sesuai dengan tujuan yang telah dirancang sejak awal.

1. *Form Input URL Berita*

Form Input URL Berita merupakan fitur yang memungkinkan pengguna memasukkan tautan berita yang ingin dianalisis oleh sistem. *Form* ini berfungsi sebagai titik awal dalam proses ekstraksi informasi dari berita *online*.

Ketika pengguna memasukkan URL berita, sistem akan mengambil konten dari halaman tersebut, termasuk judul, isi berita, serta *metadata* yang relevan. Data yang diekstrak kemudian diproses lebih lanjut untuk mendeteksi entitas seperti *event*, lokasi, dan waktu. Proses ini dilakukan secara otomatis untuk memastikan bahwa berita yang diinput dapat dianalisis dengan akurat sesuai dengan model yang telah dilatih sebelumnya.



Gambar 4.3 *Form Input URL* 3

2. *Form Hasil Scraping Berita*

Form Hasil Scraping Berita menampilkan informasi yang telah diekstrak dari URL berita yang dimasukkan oleh pengguna. Setelah sistem melakukan *scraping*, *form* ini akan menampilkan hasil ekstraksi seperti judul berita, isi berita, serta *metadata* lainnya yang berhasil diambil dari halaman sumber. Data yang ditampilkan pada *form* ini merupakan hasil pemrosesan awal sebelum dilakukan analisis lebih lanjut, seperti deteksi entitas *event*, *location*, dan waktu. Dengan adanya *form* ini, pengguna dapat melihat apakah sistem berhasil mengambil data dengan benar sebelum melanjutkan ke tahap analisis lebih lanjut.

Gambar 4.4 Tampilan Hasil *Scraping* URL 4

3. *Form* Deteksi Entitas

Form Deteksi Entitas menampilkan hasil analisis terhadap teks berita yang telah diambil melalui proses *scraping*. Pada *form* ini, sistem akan menyoroti entitas yang terdeteksi dalam teks berita, seperti *event* (peristiwa), *loc*, dan *time*. Setelah sistem melakukan pemrosesan dengan model *Named Entity Recognition* (NER), entitas yang ditemukan akan ditampilkan dalam format yang lebih mudah dipahami, misalnya dengan pemberian warna atau label pada setiap entitas yang terdeteksi. *Form* ini memungkinkan pengguna untuk melihat secara langsung hasil ekstraksi informasi yang dilakukan oleh sistem.

Entitas Terdeteksi

EVENT 8 Entitas

kebakaran tengah ludes dilalap api terjadi

kebakaran sebelumnya ruko kebakaran korban jiwa

memandamkan

LOC 10 Entitas

simeulue kebakaran aie kecamatan simeulue kabupaten simeulue

kecamatan simeulue simeulue daerah bpbd simeulue zulfadli

desa kampung aie serambinewscom tersebut aie simeulue

TIME 5 Entitas

minggu 26/2/2025 dini minggu 16/2/2025 dini hari 48

minggu 16/2/2025 dini hari 16/2/2025

Gambar 4.5 *Form Deteksi Entitas 5*

4.6 Pembahasan

4.6.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini berupa artikel berita yang diambil dari berbagai situs web yang membahas mengenai peristiwa yang terjadi di Kabupaten Simeulue. Proses pengumpulan data dilakukan dengan mencari berita menggunakan kata kunci yang relevan, seperti kebakaran, banjir, longsor, gempa bumi, pembunuhan, kasus jinayat, dan topik lain yang berkaitan dengan peristiwa di wilayah tersebut. Pencarian berita dimulai dengan menggunakan mesin pencari untuk mendapatkan tautan dari berbagai situs berita yang membahas kejadian di Kabupaten Simeulue. Setelah itu, tautan dari situs web yang relevan dikumpulkan dan dilakukan proses scraping menggunakan modul *Newspaper* dalam *Python*. Namun, tidak semua situs web berhasil di-*scraping* menggunakan modul ini, sehingga terdapat beberapa artikel yang harus dipilih secara manual berdasarkan dua pertimbangan utama, yaitu kesesuaian dengan kriteria peristiwa dan keberhasilan dalam proses *scraping*. Artikel yang dipilih harus membahas suatu kejadian atau peristiwa yang relevan dengan fokus penelitian, seperti bencana alam, tindak kriminal, atau kasus jinayat. Selain itu, artikel yang berhasil diambil

menggunakan modul *Newspaper* lebih diutamakan karena memungkinkan ekstraksi data yang lebih sistematis.

Namun, jika suatu situs web tidak dapat di-*scraping*, pemilihan artikel dilakukan secara manual dengan mempertimbangkan variasi sumber berita. Setelah proses seleksi dan *scraping* dilakukan, total berita yang berhasil dikumpulkan adalah 79 artikel. Sumber berita yang digunakan berasal dari berbagai situs web terpercaya, di antaranya *detik.com*, *kompas.com*, *liputan6.com*, *sumut.suara.com*, *tvonenews.com*, *tribatanewssimeulue.com*, *popularitas.com*, *aceh.inews.id*, *aceh.tribunnews.com*, dan *m.jpnn.com*. Berita-berita yang terkumpul kemudian diolah dan digunakan dalam proses pelatihan model *Named Entity Recognition* (NER) untuk mendeteksi entitas peristiwa berdasarkan teks berita.

4.6.2 Proses *Preprocessing Data*

Setelah data berhasil dikumpulkan, tahap selanjutnya adalah melakukan *preprocessing*, proses *preprocessing* yang dilakukan sebagai berikut:

1. Penghapusan URL

Dalam gambar 4.1 berikut menunjukkan kode implementasi untuk proses penghapusan URL yang tidak diperlukan pada penelitian ini.

```
def remove_urls(text):
    return re.sub(r'http\S+|www.\S+', '', text)
```

2. *Case Folding*

Proses *preprocessing* selanjutnya yaitu *case folding*, gambar berikut menunjukkan implementasi kode untuk proses *case folding*.

```
def case_folding(text):
    text = text.lower()
    text = re.sub(r'^a-z0-9./()]+', '', text)
    return text
```

3. Tokenisasi

Proses tokenisasi diawali dengan memecah teks berita menjadi token kata menggunakan metode *text_to_word_sequence*. Metode ini berfungsi untuk menghilangkan tanda baca dan karakter khusus yang tidak diperlukan, sehingga setiap kata dalam teks dapat dipisahkan dengan lebih terstruktur. Setelah dilakukan tokenisasi, setiap kata dalam teks akan dianggap sebagai satu token yang kemudian dihitung jumlahnya. Hasil dari proses ini menghasilkan sebanyak 20.782 token. Token-token yang diperoleh kemudian disimpan dalam dokumen berformat CSV untuk mempermudah proses analisis lebih lanjut. Penyimpanan dalam format CSV dilakukan dengan menuliskan setiap token dalam baris yang terpisah, sehingga dapat dengan mudah diakses dan digunakan dalam pelatihan model *Named Entity Recognition* (NER) atau analisis lainnya.

```
def tokenize(text):
    return text_to_word_sequence(text,
    filters='!"#$%&()*+,-.:;<=>?@[\\]^_`{|}~\t\n')
```

4.6.3 Pelebelan Data

Setelah data dipecah menjadi token, setiap token kemudian diberi label sesuai dengan kategorinya. *Labeling* ini bertujuan untuk mengklasifikasikan setiap token berdasarkan jenis entitas yang ingin dikenali, seperti *event* (peristiwa), *loc* (lokasi), dan *time* (waktu). Proses pemberian label ini menghasilkan *dataset* yang siap digunakan untuk pelatihan model seperti yang ditunjukkan pada Gambar 4.4.

	A	B	C	D
513				
514		Sebuah	O	
515		jembatan	O	
516		di	O	
517		Desa	LOC	
518		Laure'e	LOC	
519		Kecamatan	O	
520		Simeulue	LOC	
521		Tengah	LOC	
522		,	O	
523		Kabupaten	LOC	
524		Simeulue	LOC	
525		,	O	
526		Aceh	O	
527		,	O	
528		ambruk	EVENT	
529		diterjang	EVENT	
530		banjir	EVENT	
531		,	O	
532		Senin	TIME	
533		(17/5)	TIME	
534		.	O	
535				
536		Ambruknya	EVENT	
537		jembatan	EVENT	
538		tersebut	O	
539		menyebabkan	O	
540		akses	O	
541		jalan	O	
542		menyebabkan	O	

Gambar 4.6 Dataset 6

Selanjutnya, untuk menganalisis distribusi token dalam setiap kategori, jumlah token pada masing-masing label dihitung dan disajikan dalam Gambar 4.5. Hasil analisis menunjukkan bahwa dataset ini tidak seimbang, di mana label "O" (*Outside*) jauh lebih banyak dibandingkan dengan label lainnya. Label "O" menandakan bahwa token tersebut bukan bagian dari entitas yang sedang dipelajari. Ketidakseimbangan ini perlu diperhatikan dalam proses pelatihan model, karena model dapat cenderung lebih fokus pada label yang dominan, sehingga strategi seperti *resampling* atau penyesuaian bobot loss function mungkin diperlukan untuk meningkatkan performa model dalam mengenali entitas dengan lebih akurat.

4.6.4 Data Masukkan Model

Sebelum digunakan sebagai data masukan, *dataset* terlebih dahulu dikonversi ke dalam bentuk daftar yang berisi daftar lain yang terdiri dari indeks kata dan label. Proses ini dilakukan untuk memisahkan setiap kalimat dalam *dataset* sehingga

dapat dikenali sebagai unit yang berbeda. Setiap kata dalam kalimat diberikan indeks numerik melalui *word_vocab*, sedangkan labelnya dikodekan dengan indeks tertentu menggunakan *label_vocab*. Langkah ini bertujuan untuk mengubah data teks menjadi representasi angka yang dapat dipahami oleh model.

Agar model *deep learning* dapat memproses data dengan optimal, setiap sampel harus memiliki panjang yang seragam. Dalam penelitian ini, panjang maksimum kalimat ditetapkan sebanyak 512 token. Untuk mencapai panjang yang seragam, digunakan fungsi *pad_sequences()*, yang berfungsi untuk memotong kalimat yang melebihi batas panjang maksimum serta menambahkan nilai 0 (*padding*) pada kalimat yang lebih pendek. Padding ini dilakukan di bagian akhir kalimat agar struktur aslinya tetap terjaga.

Setelah proses *padding* selesai, label yang telah dikonversi menjadi angka kemudian diubah ke dalam bentuk *one-hot encoding*. Konversi ini bertujuan agar setiap label direpresentasikan dalam bentuk vektor biner yang sesuai dengan jumlah kategori label dalam dataset. Dengan cara ini, model dapat membedakan setiap label tanpa menganggapnya memiliki hubungan numerik tertentu.

```
word_vocab = {"<PAD>": 0, "<UNK>": 1}
label_vocab = {"O": 0}
for sentence in sentences:
    for word in sentence:
        if word not in word_vocab:
            word_vocab[word] = len(word_vocab)
for label_seq in labels:
    for label in label_seq:
        if label not in label_vocab:
            label_vocab[label] = len(label_vocab)
sentences_idx = [[word_vocab.get(word, word_vocab["<UNK>"])
for word in sentence] for sentence in sentences]
labels_idx = [[label_vocab[label] for label in label_seq]
for label_seq in labels]
max_seq_len = 512
```

```

padded_sentences = pad_sequences(sentences_idx,
maxlen=max_seq_len, padding="post", value=0)
padded_labels = pad_sequences(labels_idx,
maxlen=max_seq_len, padding="post", value=0)
num_classes = len(label_vocab)
categorical_labels = to_categorical(padded_labels,
num_classes=num_classes)

```

4.6.5 Feature Extraction

Pada tahap ini, digunakan teknik *word embedding* untuk mengubah setiap kata dalam *dataset* menjadi vektor numerik dengan dimensi tertentu. *Word embedding* bertujuan untuk menangkap hubungan semantik antar kata sehingga model dapat memahami makna kata berdasarkan konteks penggunaannya.

```

fasttext_model=gensim.models.KeyedVectors.load_word2vec_for
mat('cc.id.300.vec', binary=False)
embedding_dim = fasttext_model.vector_size
embedding_matrix = np.random.uniform(-0.01, 0.01,
(len(word_vocab), embedding_dim))
for word, i in word_vocab.items():
    if word in fasttext_model.key_to_index:
        embedding_matrix[i] = fasttext_model[word]

```

Dalam penelitian ini, digunakan model *FastText pretrained embedding* yang telah dilatih sebelumnya pada korpus teks dalam bahasa Indonesia. Model ini dimuat menggunakan fungsi `gensim.models.KeyedVectors.load_word2vec_format()`, dengan parameter 'cc.id.300.vec' yang merupakan *file* vektor kata berukuran 300 dimensi. Setelah model dimuat, ukuran dimensi *embedding* diambil dari properti *vector_size* milik *FastText*.

Selanjutnya, dibuat sebuah *embedding matrix* untuk menyimpan representasi vektor dari setiap kata dalam *word_vocab*. Pada awalnya, matriks ini diinisialisasi dengan nilai acak kecil menggunakan fungsi `np.random.uniform()`, yang

memberikan distribusi nilai antara -0.01 hingga 0.01. Hal ini dilakukan untuk kata-kata yang tidak ditemukan dalam model *FastText*.

Kemudian, setiap kata dalam *word_vocab* dicek apakah terdapat dalam model *FastText* menggunakan *fasttext_model.key_to_index*. Jika kata ditemukan, vektornya diambil dari model *FastText* dan disimpan di posisi indeks yang sesuai dalam *embedding_matrix*. Proses ini memastikan bahwa setiap kata memiliki representasi vektor yang sesuai dengan distribusi semantik yang telah dipelajari sebelumnya, sehingga membantu model NER dalam memahami hubungan antar kata dengan lebih baik.

Berikut ini merupakan hasil ekstraksi fitur dari 5 dimensi pertama *vector* yang diambil dengan menggunakan *word embedding* dari *word2vec fasttext*.

Tabel 4.1 Hasil Ekstraksi Fitur

Kata	Dimensi Vektor				
	1	2	3	4	5
Korban	-0.0075	0.1138	-0.0082	0.1104	-0.035
Insiden	0.0077	0.0323	-0.0019	0.0878	0.0577
Lalu	0.0237	0.013	0.0052	0.1364	0.0532
Lintas	0.017	0.0854	0.0871	0.069	0.1608
Di	-0.0048	-0.0036	-0.0809	0.4014	0.0801
Desa	-0.0568	0.0214	0.0463	0.1178	-0.0514
Suak	-0.2094	-0.0113	-0.063	0.0599	0.051
Buluh	-0.0597	0.0068	-0.0249	0.1983	-0.1187
Kecamatan	-0.0188	0.0158	0.0093	0.0661	0.0039
Simeulue	-0.1274	0.0131	-0.0185	0.0621	-0.0121
Timur	-0.0666	-0.0153	0.0115	0.1094	-0.0122
Kabupaten	-0.015	0.0235	-0.0129	0.1416	-0.0202
Pada	0.0119	0.0196	-0.016	0.1696	-0.021
Hari	-0.0277	0.0034	-0.0838	0.1497	0.0456
Kamis	-0.124	0.0294	-0.0493	0.0814	0.0147
(31/07/24)	-0.0037	-0.0013	-0.0081	-0.0082	-0.0043
Segera	-0.0083	-0.0249	0.0034	0.1078	-0.0001

Tabel 4.1 Hasil Ekstraksi Fitur (Lanjutan)

Kata	Dimensi Vektor				
	1	2	3	4	5
Mendapatkan	0.0181	0.0215	0.0215	-0.0216	-0.027
Pertolongan	-0.0146	0.0061	0.0655	0.0264	0.0093
Dari	0.0488	0.0038	0.0593	0.1177	-0.0503
Personel	0.0095	0.0285	0.0202	0.2439	-0.0514
Yang	0.0128	-0.027	0.038	0.244	-0.0289
Langsung	0.0203	0.006	0.003	0.0573	0.0036
Bergerak	-0.0113	0.0253	0.0138	0.0412	0.0213
Ke	-0.03	0.171	0.1221	0.2597	0.0149
Lokasi	-0.0381	0.0565	0.011	0.0472	-0.0063
Kejadian	0.0102	0.0058	-0.0001	0.0723	0.0384

4.6.6 Membangun Model NER

1. Arsitektur Model

Pada tahap ini, model *Named Entity Recognition* (NER) dibangun dengan menggunakan kombinasi arsitektur BiLSTM dan CNNs. Model diawali dengan *embedding layer*, yang menerima jumlah kata unik dalam kosakata ($input_dim=len(word_vocab)$) dan dimensi vektor dari *embedding FastText* ($output_dim=embedding_dim$). *Embedding* ini menggunakan *pretrained embedding FastText* yang telah dimuat sebelumnya, diinisialisasi dengan $embeddings_initializer=tf.keras.initializers.Constant(embedding_matrix)$, serta ditetapkan sebagai tidak dapat dilatih ($trainable=False$) agar mempertahankan informasi semantik dari model *FastText*. Untuk menghindari model belajar dari padding dalam sekuens, digunakan *masking layer* ($Masking(mask_value=0.0)$).

Selanjutnya, BiLSTM dengan 128 unit digunakan untuk memproses teks dari dua arah sehingga model mampu menangkap konteks sebelum dan sesudah kata dalam satu kalimat. Setelah itu, lapisan *Conv1D* diterapkan dengan 128 *filter* dan *kernel size* 3 untuk menangkap pola lokal dalam teks. Lapisan konvolusi ini diikuti oleh *MaxPooling1D*, yang berfungsi untuk mengurangi dimensi fitur dan

meningkatkan efisiensi komputasi. Model kemudian menerapkan *TimeDistributed Dense* dengan 64 unit dan aktivasi *ReLU*, yang memungkinkan setiap token dalam sekuens memiliki representasi sendiri sebelum masuk ke lapisan *dropout* (0.5) guna mencegah overfitting. Akhirnya, model memiliki lapisan keluaran dengan aktivasi *softmax*, yang mengklasifikasikan setiap kata dalam teks ke dalam kategori entitas yang sesuai.

```
model = Sequential()
model.add(Embedding(
    input_dim=len(word_vocab),
    output_dim=embedding_dim,
    embeddings_initializer=tf.keras.initializers.Constant(
embedding_matrix),
    input_length=max_seq_len,
    trainable=False,
    mask_zero=True
))

model.add(Masking(mask_value=0.0))
model.add(Bidirectional(LSTM(128, return_sequences=True)))
model.add(Conv1D(filters=128, kernel_size=3,
activation="relu", padding="same"))
model.add(MaxPooling1D(pool_size=1))
model.add(TimeDistributed(Dense(64, activation="relu")))
model.add(Dropout(0.5))
model.add(TimeDistributed(Dense(num_classes,
activation="softmax"))))

dummy_input = np.zeros((1, max_seq_len))
model(dummy_input)
print(model.summary())
```

2. Kompilasi Model

Setelah arsitektur model ditentukan, langkah selanjutnya adalah kompilasi model menggunakan *optimizer Adam*, yang terkenal stabil dalam pelatihan model

deep learning. Fungsi *loss* yang digunakan adalah *categorical_crossentropy*, karena tugas ini merupakan klasifikasi multi-kelas dengan *output* yang terdistribusi dalam bentuk *one-hot encoding*. Selain itu, metrik evaluasi yang digunakan adalah akurasi, untuk mengukur sejauh mana model dapat memprediksi label dengan benar.

Salah satu tantangan dalam pelatihan model NER adalah ketidakseimbangan jumlah sampel untuk setiap label, yang dapat menyebabkan bias model terhadap label mayoritas. Untuk mengatasi hal ini, dilakukan perhitungan *class weight* menggunakan *compute_class_weight('balanced')*. Fungsi ini menghitung bobot kelas berdasarkan distribusi label dalam *dataset*, sehingga kelas yang lebih jarang mendapatkan bobot lebih besar agar model tidak mengabaikan entitas yang lebih jarang muncul. Hasil perhitungan *class weight* kemudian disimpan dalam bentuk dictionary (*class_weights_dict*), di mana setiap indeks label memiliki bobot yang sesuai.

```
model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])
```

3. Splitting Data

Setelah model dikompilasi, dataset dibagi menjadi 80% data latih dan 20% data uji menggunakan *train_test_split()*, dengan parameter *test_size=0.2* untuk memastikan proporsi yang seimbang antara data latih dan uji. *Dataset* yang telah diproses sebelumnya (*padded_sentences* dan *categorical_labels*) digunakan sebagai input dan label dalam proses pembagian ini.

Untuk mengimbangi distribusi label dalam data latih, dilakukan penyesuaian bobot sampel melalui *sample weights*. *Sample weights* dibuat dalam bentuk *array* dengan ukuran yang sama seperti label (*sample_weights = np.zeros((train_labels.shape[0], train_labels.shape[1]))*). Setiap label dalam data latih diperiksa, dan bobotnya ditentukan berdasarkan *class weight* yang telah dihitung sebelumnya. Dengan cara ini, model dapat memberikan perhatian lebih terhadap label yang kurang dominan, sehingga meningkatkan akurasi dalam mengenali entitas yang jarang muncul.

```
train_sentences, test_sentences, train_labels, test_labels = train_test_split(
    padded_sentences, categorical_labels, test_size=0.2,
    random_state=42
)
```

```

sample_weights = np.zeros((train_labels.shape[0],
train_labels.shape[1]))
for i, seq in enumerate(train_labels):
    for j in range(len(seq)):
        label_index = np.argmax(seq[j])
        sample_weights[i, j] =
class_weights_dict.get(label_index, 1.0)

```

4. Melatih Model

Model dilatih menggunakan *dataset* yang telah disiapkan dengan menjalankan fungsi *model.fit()*. Model dilatih selama 50 *epoch*, menggunakan *batch size* 32 untuk memastikan proses pelatihan berjalan dengan optimal tanpa memakan terlalu banyak memori. Data latih yang digunakan adalah *train_sentences* dan *train_labels*, dengan *validation split* sebesar 20%, yang berarti sebagian dari data latih akan digunakan sebagai data validasi guna memantau kinerja model selama pelatihan berlangsung.

Selain itu, parameter *sample_weight* diterapkan pada saat pelatihan untuk menyesuaikan bobot dari setiap sampel berdasarkan distribusi label dalam *dataset*. Hal ini bertujuan agar model tidak hanya belajar dari label mayoritas, tetapi juga mampu mengenali entitas yang kurang sering muncul dalam *dataset*. Dengan pendekatan ini, model NER yang dihasilkan diharapkan memiliki generalisasi yang baik dalam mendeteksi berbagai entitas pada teks berita.

```

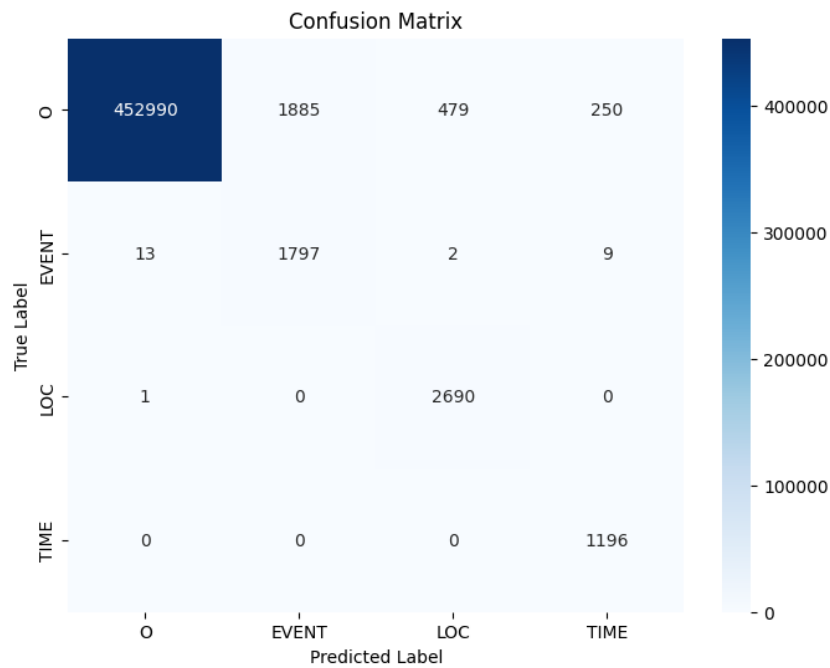
history = model.fit(train_sentences, train_labels,
epochs=50, validation_split=0.2, batch_size=32,
sample_weight=sample_weights)

```

4.6.7 Evaluasi Model

Pada penelitian ini dilakukan evaluasi model untuk mengukur seberapa baik model dalam melakukan tugas *Named Entity Recognition (NER)* dengan mengklasifikasikan entitas dalam teks ke dalam kategori *EVENT*, *LOC*, *TIME*, atau *O* (Outside). Evaluasi ini menggunakan data uji yang telah diproses dan dianalisis menggunakan *confusion matrix*. *Confusion matrix* memberikan gambaran tentang jumlah prediksi yang benar dan salah yang dibuat oleh model dalam mengklasifikasikan setiap entitas. Dari hasil evaluasi, model menunjukkan performa yang sangat baik dalam mengenali kelas *O* dengan jumlah prediksi benar

yang dominan. Namun, terdapat beberapa kesalahan klasifikasi, terutama pada kelas *EVENT* yang terkadang salah diklasifikasikan sebagai *O* atau kelas lain. Sementara itu, kelas *LOC* dan *TIME* memiliki akurasi yang lebih baik dengan sedikit kesalahan klasifikasi. Evaluasi ini membantu dalam memahami kekuatan dan kelemahan model, serta memberikan wawasan untuk meningkatkan akurasi dengan pendekatan seperti *data balancing* atau penggunaan *weighted loss function*.



Gambar 4.7 Evaluasi *Confusion Matrix* 7

Dari *confusion matrix* di atas, dapat diukur *F1-Score* untuk setiap label seperti berikut:

1. *F1-Score* untuk label O

$$Precision\ O = \frac{452990}{452990+2614} = \frac{452990}{455604} = 0.994$$

$$Recall\ O = \frac{452990}{452990+14} = \frac{452990}{453004} = 0.999$$

$$F1\ O = 2 \times \frac{0.994 \times 0.999}{0.994 + 0.999} = 0.996$$

2. *F1-Score* untuk label Event

$$Precision\ Event = \frac{1797}{1797+13} = \frac{1797}{1810} = 0.993$$

$$Recall\ Event = \frac{1797}{1797+1885} = \frac{1797}{3682} = 0.488$$

$$F1\ Event = 2 \times \frac{0.993 \times 0.488}{0.993 + 0.488} = 0.654$$

3. F1-Score untuk label Loc

$$Precision\ Loc = \frac{2690}{2690+2} = \frac{2690}{2692} = 0.999$$

$$Recall\ Loc = \frac{2690}{2690+479} = \frac{2690}{3169} = 0.849$$

$$F1\ Loc = 2 \times \frac{0.999 \times 0.849}{0.999 + 0.849} = 0.918$$

4. F1-Score untuk label Time

$$Precision\ Time = \frac{2690}{2690+2} = \frac{2690}{2692} = 0.999$$

$$Recall\ Time = \frac{2690}{2690+479} = \frac{2690}{3169} = 0.849$$

$$F1\ Time = 2 \times \frac{0.999 \times 0.849}{0.999 + 0.849} = 0.902$$

Dengan demikian, perhitungan rata-rata F1-Score adalah:

$$Macro\ F1 - Score = \frac{1}{4} (0.996 + 0.654 + 0.918 + 0.902) = \frac{1}{4} (3.47) = 0.8675$$

Jadi, rata-rata F1-Score untuk model ini adalah 86.75%. Ini menunjukkan bahwa secara keseluruhan, model memiliki keseimbangan yang cukup baik antara Precision dan Recall pada semua kelas, meskipun performa pada kelas EVENT lebih rendah dibandingkan kelas lainnya.


Evaluasi Model:				
	precision	recall	f1-score	support
O	1.00	0.99	1.00	455604
EVENT	0.49	0.99	0.65	1821
LOC	0.85	1.00	0.92	2691
TIME	0.82	1.00	0.90	1196
accuracy			0.99	461312
macro avg	0.79	1.00	0.87	461312
weighted avg	1.00	0.99	1.00	461312

Gambar 4.7 Hasil Evaluasi 8



4.6.8 Pengujian Sistem dan Deteksi Entitas

Pada tahap ini, sistem deteksi entitas diimplementasikan menggunakan *Flask*, yang merupakan *framework* web berbasis *Python* yang ringan dan mudah digunakan. *Flask* dipilih karena fleksibilitas dan kemampuannya dalam mengintegrasikan model *deep learning* ke dalam aplikasi web dengan cepat. Sebelum diimplementasikan, model yang digunakan untuk deteksi entitas terlebih dahulu disimpan dalam beberapa *file* penting. Pertama, *bilstm_cnn_ner.keras*, yang menyimpan arsitektur dan bobot model BiLSTM-CNN yang digunakan untuk *Named Entity Recognition* (NER). Kedua, *word_vocab.pkl*, yang berisi kamus (*vocabulary*) kata yang digunakan untuk mengonversi kata menjadi indeks numerik sebelum dimasukkan ke dalam model. Ketiga, *label_vocab.pkl*, yang berisi kamus label yang digunakan untuk mengonversi label entitas menjadi indeks numerik dan sebaliknya. Terakhir, *embedding_matrix.npy*, yang menyimpan *embedding matrix* untuk merepresentasikan kata dalam bentuk vektor numerik berdimensi tinggi. Model disimpan dalam *file* terpisah untuk memudahkan proses *load* dan *inference* saat diintegrasikan ke dalam aplikasi *Flask*.

Tabel 4.2 Pengujian Dan Deteksi Entitas 1

No	Pengujian	Uji Coba	Hasil Pengujian	Ket
1	Form Input URL	Klik Button Scrap		Valid

Tabel 4.2 Pengujian dan Deteksi Entitas (Lanjutan)

No	Pengujian	Uji Coba	Hasil Pengujian	Ket
2	Form Hasil Scraping Berita	Hasil Scraping		Valid
3	Form Deteksi	Klik Button Deteksi Entitas		Valid

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dengan melakukan implementasi dan pengujian model, dapat ditarik Kesimpulan bahwa:

1. Kombinasi Algoritma *Deeplearning* BiLSTM dan CNNs terbukti efektif dalam mendeteksi entitas dalam berita dengan memisahkan kategori *loc*, *event* dan *time*. Proses pengembangan sistem mencakup tahapan pengumpulan data, *preprocessing*, ekstraksi fitur menggunakan *Word2Vec* dari *Fasttext*, pelatihan model, dan evaluasi performa model.
2. Sistem ini dibangun dengan memanfaatkan 20.782 token dari 941 kalimat pada 100 berita yang diberi label *loc*, *event* dan *time*. Hasil evaluasi terhadap model BiLSTM - CNNs menunjukkan bahwa sistem ini berhasil mencapai keseluruhan akurasi sebesar 99%. Nilai precision pada entitas *event* 49%, recall 99%, dan F1-score 65%. Pada entitas *loc* nilai precision mencapai 85%, recall 100% dan F1-score 92%. Sedangkan pada entitas *time* nilai precision mencapai 82%, recall 100% dan F1-score 90%. Secara keseluruhan rata – rata F1-score adalah 86.75%. Tingginya nilai akurasi mengindikasikan bahwa model mampu secara konsisten mengklasifikasikan entitas berita dengan benar sebagai *loc*, *event* dan *time*.

5.2 Saran

Berdasarkan hasil yang dicapai dalam penelitian ini, beberapa saran dapat diajukan untuk penelitian selanjutnya:

1. Pengembangan model dapat dikombinasikan dengan menggunakan metode lain seperti BERT, IndoBERT, CRF dan lain sebagainya agar dapat meningkatkan kinerja model yang lebih optimal dan akurat.
2. Mengumpulkan lebih banyak data untuk memperbaiki keseimbangan antara label sangat dianjurkan. Teknik augmentasi data dapat diterapkan untuk meningkatkan jumlah data yang tersedia dan memberikan representasi yang lebih baik dari berbagai tipe label.

DAFTAR PUSTAKA

- Azizi, M. R., Hayuhardhika, W., Putra, N., & Arwani, I. (2023). Ekstraksi Informasi pada Data Logbook KKN Mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya Malang menggunakan Metode NER (Named Entity Recognition). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 7(6), 2895–2903. <http://j-ptiik.ub.ac.id>
- Batbaatar, E., & Ryu, K. H. (2019). Ontology-based healthcare named entity recognition from twitter messages using a recurrent neural network approach. *International Journal of Environmental Research and Public Health*, 16(19), 1–19. <https://doi.org/10.3390/ijerph16193628>
- Djiwadikusumah, F., Irawan, G. H., & Haekal, R. (2021). Web Scraping Situs E-Commerce Menggunakan Teknik Parsing Dom. *Sains Dan Teknologi*, 7(2), 52–57.
- Fudholi, D. H., Nayoan, R. A. N., Hidayatullah, A. F., & Arianto, D. B. (2022). a Hybrid Cnn-Bilstm Model for Drug Named Entity Recognition. *Journal of Engineering Science and Technology*, 17(1), 730–744.
- Hidayatullah, A. F., Putra, M. F. D. A., Wibowo, A. P., & Nastiti, K. R. (2023). Named Entity Recognition on Tourist Destinations Reviews in the Indonesian Language. *Jurnal Linguistik Komputasional (JLK)*, 6(1), 30–35. <https://doi.org/10.26418/jlk.v6i1.89>
- Khder, M. A. (2021). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing and Its Applications*, 13(3), 144–168. <https://doi.org/10.15849/ijasca.211128.11>
- Li, C., Wang, Z., Rao, M., Belkin, D., Song, W., Jiang, H., Yan, P., Li, Y., Lin, P., Hu, M., Ge, N., Strachan, J. P., Barnell, M., Wu, Q., Williams, R. S., Yang, J. J., & Xia, Q. (2019). Long short-term memory networks in memristor crossbar arrays. *Nature Machine Intelligence*, 1(1), 49–57. <https://doi.org/10.1038/s42256-018-0001-4>
- Liang, M., & Shi, Y. (2023). Named Entity Recognition Method Based on BERT-whitening and Dynamic Fusion Model. *2023 5th International Conference on Natural Language Processing (ICNLP)*, 191–197. <https://doi.org/10.1109/ICNLP58431.2023.00041>
- Lin, J., & Liu, E. (2022). Research on Named Entity Recognition Method of Metro On-Board Equipment Based on Multiheaded Self-Attention Mechanism and CNN-BiLSTM-CRF. *Computational Intelligence and Neuroscience*, 2022. <https://doi.org/10.1155/2022/6374988>
- Lindemann, B., Müller, T., Vietz, H., Jazdi, N., & Weyrich, M. (2021). A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, 99(July 2020), 650–655. <https://doi.org/10.1016/j.procir.2021.03.088>
- Liu, P., Sun, Z., & Zhou, B. (2024). An ELECTRA-Based Model for Power Safety

- Named Entity Recognition. *Applied Sciences*, 14(20), 9410. <https://doi.org/10.3390/app14209410>
- Lv, Y., Qin, X., Du, X., & Qiu, S. (2023). Deep adaptation of CNN in Chinese named entity recognition. *Engineering Reports*, 5(6), 1–14. <https://doi.org/10.1002/eng2.12614>
- Permana, H., Purnamasari, K. K., Dipati, J., No, U., Bandung, K., & Barat, J. (2022). Named Entity Recognition Menggunakan Metode Bidirectional Lstm-Crf Pada Teks Bahasa Indonesia. *Universitas Komputer Indonesia*, 112.
- Putra, A. A., & Kurniawan, R. (2021). Bidirectional Lstm-Cnns Untuk Ekstraksi Entity Lokasi Kebakaran Pada Berita Online Berbahasa Indonesia. *Seminar Nasional Official Statistics*, 2020(1), 319–327. <https://doi.org/10.34123/semnasoffstat.v2020i1.601>
- Rifani, R., Bijaksana, M. A., & Asror Ibnu. (2019). Named Entity Recognition for an Indonesian Based Language Tweet using Multinomial Naive Bayes Classifier. *Ind. Journal on Computing*, 4(2), 119–126. <https://doi.org/10.21108/indoic.2019.4.2.330>
- Santoso, J., Setiawan, E. I., Purwanto, C. N., Yuniarno, E. M., Hariadi, M., & Purnomo, M. H. (2021). Named entity recognition for extracting concept in ontology building on Indonesian language using end-to-end bidirectional long short term memory. *Expert Systems with Applications*, 176(March), 114856. <https://doi.org/10.1016/j.eswa.2021.114856>
- Shidik, G. F., Saputra, F. O., Saraswati, G. W., Winarsih, N. A. S., Rohman, M. S., Pramunendar, R. A., Kusuma, E. J., Ratmana, D. O., Venus, V., Andono, P. N., & Hasibuan, Z. A. (2024). Indonesian disaster named entity recognition from multi source information using bidirectional LSTM (BiLSTM). *Journal of Open Innovation: Technology, Market, and Complexity*, 10(3), 100358. <https://doi.org/10.1016/j.joitmc.2024.100358>
- Shiraishi, E., de Camargo, R. Y., Silva, H. L. P., & Prati, R. C. (2024). Retrieval-Enhanced Named Entity Recognition. *Cornell University*. <https://doi.org/https://doi.org/10.48550/arXiv.2410.13118>
- Subowo, E., Adi Artanto, F., Putri, I., & Umaedi, W. (2022). Algoritma Bidirectional Long Short Term Memory untuk Analisis Sentimen Berbasis Aspek pada Aplikasi Belanja Online dengan Cicilan. *Jurnal Fasilkom*, 12(2), 132–140. <https://doi.org/10.37859/jf.v12i2.3759>
- Sukardi, S., Susanty, M., Irawan, A., & Randi Fermana Putra. (2021). Low Complexity Named-Entity Recognition for Indonesian Language using BiLSTM-CNNs. *IEEE*. <https://doi.org/https://doi.org/10.1109/ICOIACT50329.2020.9331989>
- Sun, Z., & Li, X. (2023). Named Entity Recognition Model Based on Feature Fusion. *Information*, 14(2), 133. <https://doi.org/10.3390/info14020133>
- Theofany, M., Anwar, A., Wijoyo, S. H., & Nugraha, W. H. (2024). Implementasi Metode Textrank Dan Named Entity Recognition Untuk Ekstraksi Kata Kunci

Pada Media Online Berita. *Jurnal Sistem Informasi, Teknologi Informasi, Dan Edukasi Sistem Informasi*, 5(1), 34–41.

- Tjut Adek, R., Kesuma Dinata, R., & Ditha, A. (2021). Online Newspaper Clustering in Aceh using the Agglomerative Hierarchical Clustering Method. *International Journal of Engineering, Science and Information Technology*, 2(1), 70–75. <https://doi.org/10.52088/ijesty.v2i1.206>
- Widiyanti, N. F., Sukmana, H. T., Hulliyah, K., Khairani, D., & Oh, L. K. (2023). Improving Indonesian Named Entity Recognition for Domain Zakat Using Conditional Random Fields. *Jurnal Online Informatika*, 8(2), 131–138. <https://doi.org/10.15575/join.v8i2.898>
- Yan, H., Sun, Y., Li, X., & Qiu, X. (2023). An Embarrassingly Easy but Strong Baseline for Nested Named Entity Recognition. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2, 1442–1452. <https://doi.org/10.18653/v1/2023.acl-short.123>
- Zahra, A. (2021). *Pemodelan Named Entity Recognition Pada Artikel Wisata Dengan Metode Birectional Long Short-Term Memory Dan Conditional Random Fields*. Universitas Islam Indonesia.