

***COREFERENCE RESOLUTION UNTUK TEKS BAHASA INDONESIA
MENGGUNAKAN RANDOM FOREST CLASSIFIER***

SKRIPSI

NIA ULAN SARI

171402045



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

2024

***COREFERENCE RESOLUTION UNTUK TEKS BAHASA INDONESIA
MENGGUNAKAN RANDOM FOREST CLASSIFIER***

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Teknologi Informasi

**NIA ULAN SARI
171402045**



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
2024**

PERSETUJUAN

Judul : *COREFERENCE RESOLUTION UNTUK TEKS BAHASA INDONESIA MENGGUNAKAN RANDOM FOREST CLASSIFIER*

Kategori : SKRIPSI

Nama : NIA ULAN SARI

Nomor Induk Mahasiswa : 171402045

Program Studi : SARJANA (S-1) TEKNOLOGI INFORMASI

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

Medan, 11 Juli 2024

Komisi Pembimbing:

Pembimbing 2

Dr. Romi Fadillah Rahmat B.Comp.Sc., M.Sc.
NIP. 198603032010121004

Pembimbing 1

Sarah Purnamawati ST., M.Sc.
NIP. 198302262010122003

Diketahui/disetujui oleh

Program Studi S1 Teknologi Informasi

Ketua,



Dedy Arisandi, ST., M.Kom.
NIP. 197908312009121002

PERNYATAAN

*COREFERENCE RESOLUTION UNTUK TEKS BAHASA INDONESIA
MENGGUNAKAN RANDOM FOREST CLASSIFIER*

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 11 Juli 2024

Nia Ulan Sari
171402045

UCAPAN TERIMA KASIH

Puji dan syukur penulis ucapkan kepada Allah *Subhanahu Wa Ta'ala*, atas berkat, rahmat serta karunia-Nya penulis dapat menyelesaikan penyusunan skripsi ini dengan baik sebagai syarat untuk menerima gelar Sarjana Komputer, Program Studi S1 Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

Penulis menyadari bahwa penelitian skripsi ini tidak akan terwujud tanpa adanya doa serta dukungan dari berbagai pihak selama proses perkuliahan sampai dengan penyelesaian skripsi ini. Dengan kerendahan dan ketulusan hati, penulis mengucapkan terima kasih kepada:

1. Kedua orang tua penulis, Ayah Noferi dan Ibu Pipi yang senantiasa memberikan dukungan, kasih sayang serta doa yang tak pernah putus untuk penulis.
2. Arief Budi Mulia selaku adik penulis yang selalu memberikan dorongan dan semangat.
3. Bapak Dr.Muryanto Amin, S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
4. Ibu Dr. Maya Silvi Lydia, M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
5. Bapak Dedy Arisandi, ST., M.Kom. selaku Ketua Program Studi Teknologi Informasi Universitas Sumatera Utara.
6. Ibu Sarah Purnamawati ST., M.Sc. selaku Dosen Pembimbing I dan Bapak Romi Fadillah Rahmat B.Comp.Sc., M.Sc selaku Dosen Pembimbing II yang bersedia memberikan bimbingan, kritik, dan saran yang membangun kepada penulis selama proses penggerjaan skripsi.
7. Seluruh dosen di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang telah memberikan ilmu pengetahuan selama masa perkuliahan.

8. Seluruh pegawai di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang membantu urusan administrasi akademik selama perkuliahan.
9. Teman seperjuangan penulis, Eka Khairani Hutaurok, Rizki Noprianita dan Bella Savira yang bersedia menjadi tempat diskusi selama masa perkuliahan dan penggeraan skripsi.
10. Rukiah Nasution selaku teman SMA yang setia menemani hari-hari dan tempat bercerita apapun via *whatsapp chat*.
11. Teman-teman mahasiswa angkatan 2017 yang menemani dan berjuang bersama penulis dari awal masuk perkuliahan hingga selesai penyusunan skripsi.
12. Semua pihak yang berperan serta langsung maupun tidak langsung yang tidak dapat disebutkan satu per satu.

Semoga Allah *Subhanahu Wa Ta'ala* selalu memberikan berkah serta karunia-Nya kepada semua pihak yang turut membantu penulis.

Medan, 11 Juli 2024

Penulis

**COREFERENCE RESOLUTION UNTUK TEKS BAHASA INDONESIA
MENGGUNAKAN RANDOM FOREST CLASSIFIER**

ABSTRAK

Coreference Resolution adalah sebuah subtugas dalam *Natural Language Processing* (NLP) yang berfokus pada identifikasi dan penyelesaian masalah referensi dari dua atau lebih entitas yang sama dalam teks. Dalam teks bahasa Indonesia, khususnya dalam novel, *coreference resolution* menjadi krusial karena bahasa yang kompleks dan variasi entitas serta referensi yang kaya. Tokoh dan entitas dalam novel sering berinteraksi, dan referensi terhadap tokoh dapat muncul berulang kali. Permasalahan lainnya adalah adanya kata ganti kepemilikan yang banyak digunakan dalam teks novel berupa imbuhan bukan kata utuh dapat menyebabkan kebingungan dalam penentuan referensi antar entitas. Maka dari itu dibuat penelitian *coreference resolution* untuk teks bahasa Indonesia dengan pendekripsi kata ganti kepemilikan imbuhan menggunakan metode *Random Forest Classifier*. Penelitian ini juga memanfaatkan *Part-of-Speech Tagging* (POS Tag) dan *Named Entity Recognition* (NER) untuk memaksimalkan pendekripsi entitas orang. Dengan menggunakan 18 teks novel sebagai data latih dan 10 teks novel sebagai data uji setelah tahap pre-processing, total terdapat 109306 pasangan entitas dan kata ganti dalam data latih, serta 4938 pasangan dalam data uji. Penelitian ini menggunakan *RandomSearchCV* untuk membantu algoritma *Random Forest Classifier* menemukan *hyperparameter* terbaik dalam proses *training*. Dengan menggunakan metode evaluasi *confusion matrix*, nilai metriks yang didapat dari hasil pengujian seluruh data *test* adalah dengan akurasi sebesar 85,5%, presisi sebesar 85%, *recall* sebesar 82,2%, dan *f1-score* sebesar 83,6 %.

Kata kunci: *coreference resolution, random forest classifier, natural language processing, novel, bahasa indonesia, randomsearchcv*

COREFERENCE RESOLUTION FOR INDONESIAN TEXT USING RANDOM FOREST CLASSIFIER

ABSTRACT

Coreference Resolution is a subtask in Natural Language Processing (NLP) that focuses on identifying and solving the reference problem of two or more similar entities in text. In Indonesian texts, especially in novels, coreference resolution is crucial because of the complex language and rich variety of entities and references. Characters and entities in novels often interact, and references to characters may appear repeatedly. Another problem is that the presence of possessive pronouns which are widely used in novel texts in the form of affixes rather than complete words can cause confusion in determining references between entities. Therefore, coreference resolution research was carried out for Indonesian texts by detecting affix possessive pronouns using the Random Forest Classifier method. This research also utilizes Part-of-Speech Tagging (POS Tag) and Named Entity Recognition (NER) to maximize detection of person entities. By using 18 novel texts as training data and 10 novel texts as test data after the pre-processing stage, there are a total of 109306 entity and pronoun pairs in the training data, and 4938 pairs in the test data. This research uses RandomSearchCV to help the Random Forest Classifier algorithm find the best hyperparameters in the training process. By using the confusion matrix evaluation method, the metric values obtained from the test results of all test data are an accuracy of 85.5%, precision of 85%, recall of 82.2%, and f1-score of 83.6%.

Keywords: coreference resolution, random forest classifier, natural language processing, novel, indonesian, randomsearchcv

DAFTAR ISI

	Hal.
PERSETUJUAN	ii
PERNYATAAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	4
1.3. Batasan Masalah	4
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	5
1.6. Metodologi Penelitian	5
1.6.1. Studi literatur	5
1.6.2. Analisis permasalahan	5
1.6.3. Perancangan sistem	5
1.6.4. Implementasi dan pengujian sistem	5
1.6.5. Dokumentasi dan penyusunan laporan	6
1.7. Sistematika Penulisan	6
BAB 2 LANDASAN TEORI	7
2.1. Novel	7
2.2. <i>Natural Language Processing</i> (NLP)	8
2.3. <i>Pre-processing</i> (Pra-pengolahan)	8
2.3.1. <i>Filtering</i>	8
2.3.2. <i>Sentence tokenization</i>	9

2.3.3. <i>Word tokenization</i>	9
2.3.4. Pemisahan kata berimbuhan	9
2.3.5. Ekstraksi fitur	9
2.4. <i>Part-of-Speech Tagging</i> (POS Tagging)	10
2.5. <i>Named Entity Recognition</i> (NER)	12
2.6. <i>Coreference Resolution</i>	13
2.7. <i>Conditional Random Fields</i> (CRF)	14
2.8. <i>Random Forest Classifier</i>	16
2.9. Penelitian Terdahulu	18
2.10. Perbedaan Penelitian	21
BAB 3 ANALISIS DAN PERANCANGAN SISTEM	22
3.1. Data yang digunakan	22
3.2. Arsitektur Umum	24
3.3. <i>Pre-processing</i> (Pra-pengolahan)	25
3.3.1. <i>Filtering</i>	25
3.3.2. <i>Sentence tokenization</i>	26
3.3.3. <i>Word tokenization</i>	27
3.3.4. Pemisahan kata berimbuhan	28
3.3.5. <i>Part-of-speech tagging</i> (POS tagging)	31
3.3.6. <i>Named entity recognition</i>	34
3.4. Pelabelan <i>Coreference</i>	37
3.5. Ekstraksi Fitur	43
3.6. Proses <i>Training</i>	48
3.7. Metode Evaluasi	51
3.8. Perancangan Sistem	52
3.8.1. Rancangan tampilan halaman beranda	52
3.8.2. Rancangan tampilan halaman <i>training</i>	52
3.8.3. Rancangan tampilan halaman <i>testing</i>	54
BAB 4 IMPLEMENTASI DAN PENGUJIAN	57
4.1. Implementasi Sistem	57
4.1.1. Spesifikasi perangkat keras dan perangkat lunak	57
4.1.2. Implementasi perancangan tampilan antarmuka	58

4.2. Pengujian Sistem	68
4.2.1. Implementasi model <i>Random Forest Classifier</i>	68
4.2.2. Pengujian model	71
4.2.3. Evaluasi	74
BAB 5 KESIMPULAN DAN SARAN	76
5.1. Kesimpulan	76
5.2. Saran	76
DAFTAR PUSTAKA	77

DAFTAR TABEL

Tabel 2.1	Daftar <i>Tagset</i>	11
Tabel 2.2	Penelitian Terdahulu	20
Tabel 3.1	Rincian <i>dataset</i> yang digunakan	22
Tabel 3.2	Penerapan proses <i>filtering</i>	26
Tabel 3.3	Penerapan proses <i>sentence tokenization</i>	27
Tabel 3.4	Penerapan proses <i>word tokenization</i>	28
Tabel 3.5	Penerapan proses pemisahan kata berimbuhan	30
Tabel 3.6	Penerapan proses POS <i>tagging</i>	33
Tabel 3.7	Penerapan proses NER	36
Tabel 3.8	Token-token yang diberi label	38
Tabel 3.9	Data hasil pelabelan	41
Tabel 3.10	Ekstraksi fitur	44
Tabel 3.11	Penerapan proses ekstraksi fitur	46
Tabel 3.12	Data fitur dan label	49
Tabel 3.13	<i>Confusion Matrix</i>	51
Tabel 4.1	Spesifikasi perangkat keras	57
Tabel 4.2	Spesifikasi perangkat lunak	57
Tabel 4.3	Performansi <i>Hyperparameter Tuning</i>	69
Tabel 4.4	Fitur data teks novel “ <i>Refrain</i> ”	72
Tabel 4.5	Tabel hasil pengujian data <i>test</i> teks "Refrain"	72
Tabel 4.6	<i>Confusion matrix</i> tiap data <i>test</i>	74

DAFTAR GAMBAR

	Hal.
Gambar 1.1 Teks novel bahasa Indonesia	1
Gambar 2.1 Pendekatan BIO pada NER	12
Gambar 2.2 Pendekatan BILOU pada NER	13
Gambar 2.3 Ilustrasi <i>coreference</i>	14
Gambar 2.4 Struktur <i>linear-chain CRF</i> (Fanny et al. 2022)	15
Gambar 2.5 Struktur metode <i>Decision Tree</i> (Geeksforgeeks, 2023)	16
Gambar 2.6 Alur kerja metode <i>Random Forest Classifier</i> (Bernal, 2023)	17
Gambar 3.1 Teks novel Sherlock Holmes	23
Gambar 3.2 Arsitektur Umum	24
Gambar 3.3 Diagram alur proses <i>filtering</i>	25
Gambar 3.4 Diagram alur proses <i>sentence tokenization</i>	26
Gambar 3.5 Diagram alur proses <i>word tokenization</i>	27
Gambar 3.6 Diagram alur pemisahan kata berimbuhan	29
Gambar 3.7 Format <i>Indonesian Manually Tagged Corpus</i>	32
Gambar 3.8 Diagram alur proses POS <i>tagging</i>	32
Gambar 3.9 Diagram alur proses NER	35
Gambar 3.10 Format pelabelan data tunggal	40
Gambar 3.11 Format pelabelan data jamak	40
Gambar 3.12 Rancangan halaman Beranda	52
Gambar 3.13 Rancangan halaman <i>Training</i>	53
Gambar 3.14 Rancangan halaman <i>Testing</i>	55
Gambar 4.1 Tampilan halaman Beranda	59
Gambar 4.2 Tampilan awal halaman <i>Training</i>	60
Gambar 4.3 Tampilan halaman <i>Training</i> yang menampilkan tabel data teks	60
Gambar 4.4 Tampilan halaman <i>Training</i> yang menampilkan tabel hasil proses NER	61

Gambar 4.5	Tampilan halaman <i>Training</i> yang menampilkan tabel hasil ekstraksi fitur	62
Gambar 4.6	Tampilan halaman <i>Training</i> yang menampilkan rincian <i>hyperparameter</i> model	63
Gambar 4.7	Tampilan halaman <i>Training</i> yang menampilkan metriks evaluasi dan <i>confusion matrix</i>	63
Gambar 4.8	Tampilan awal halaman <i>Testing</i>	64
Gambar 4.9	Tampilan halaman <i>Testing</i> yang menampilkan tabel data teks	65
Gambar 4.10	Tampilan halaman <i>Testing</i> yang menampilkan tabel hasil ekstraksi fitur	65
Gambar 4.11	Tampilan halaman <i>Testing</i> yang menampilkan hasil prediksi <i>coreference</i>	66
Gambar 4.12	Tampilan halaman <i>Testing</i> yang menampilkan hasil evaluasi data <i>test</i> perbandingan prediksi dan label	67
Gambar 4.13	Tampilan halaman <i>Testing</i> yang menampilkan metriks evaluasi dan <i>confusion matrix</i>	67
Gambar 4.14	Evaluasi <i>Confusion Matrix</i> pada data validasi	71

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Coreference resolution merupakan subtugas dari *Natural Language Processing* (NLP) yang berkaitan dengan identifikasi dan penyelesaian masalah referensi dari dua atau lebih entitas yang sama dalam teks. *Coreference resolution* merupakan langkah penting untuk beberapa tugas NLP tingkat lanjut yang melibatkan pemahaman bahasa alami seperti peringkasan dokumen, sistem dialog, terjemahan mesin, dan ekstraksi informasi (Ming, 2020).

Dalam konteks novel berbahasa Indonesia, penelitian *coreference resolution* menjadi sangat penting. Novel merupakan karya sastra yang menggunakan bahasa yang kompleks serta kaya akan variasi entitas dan referensi. Dalam novel, tokoh dan entitas berinteraksi, dan referensi pada tokoh dapat muncul berulang kali. Oleh karena itu, analisis *coreference resolution* pada novel bahasa Indonesia dapat memberikan wawasan tentang pemahaman teks dan analisis bahasa dalam konteks sastra.

Namun, dalam *coreference resolution* pada novel bahasa Indonesia juga menghadapi tantangan tersendiri. Beberapa tantangan ini termasuk variasi referensi pada entitas dengan variasi nama atau alias, penggunaan kata ganti yang ambigu, dan keberadaan frasa idiomatik yang dapat mempengaruhi hubungan referensial. Selain itu, perbedaan gaya penulisan dan gaya bahasa dalam novel juga mempengaruhi kinerja sistem *coreference resolution*. Misalkan pada teks pada Gambar 1.1.

Bagi **Sherlock Holmes₁**, **dia₂** adalah wanita yang istimewa. **Dia₃** tak pernah menyebut wanita itu dengan istilah lain. Di matanya wanita itulah yang paling hebat di antara seluruh kaumnya₄. Ini tidak berarti bahwa **Holmes₅** mencintai **Irene Adler₆**.

Gambar 1.1 Teks novel bahasa Indonesia

Pada teks di atas terdapat dua entitas orang berbeda yaitu “Sherlock Holmes₁”, dan “Irene Adler₆”, dua kata ganti orang ketiga tunggal “dia₂” dan “Dia₃” serta imbuhan akhiran (sufiks) “nya₄” yang berperan sebagai kata ganti orang ketiga tunggal. Kata ganti “dia₂” bisa saja diinterpretasikan merujuk ke entitas “Sherlock Holmes₁” karena merupakan entitas yang mendahului (*antecedant*) kata tersebut dan berada pada kalimat yang sama tetapi faktanya “dia₂” merujuk ke “Irene Adler₆” yang berada pada kalimat keempat. Kemudian entitas “Holmess₅” bisa saja diinterpretasikan sebagai entitas yang berbeda dengan “Sherlock Holmes₁” karena perbedaan jumlah karakter huruf padahal kedua entitas tersebut merujuk ke entitas orang yang sama, yang menjadi pembeda ialah “Sherlock Holmes₁” merupakan nama lengkap sedangkan “Holmess₅” hanya nama belakang.

Sebelumnya *coreference resolution* telah banyak diteliti dalam berbagai bahasa seperti bahasa Korea, Cina, Inggris, dan Indonesia. Penelitian pada bahasa Korea dilakukan oleh Seok-Won *et al.* (2016) menggunakan metode *Random Forest* dengan *sieve-guided features* menggunakan data berita bisbol. Penelitian ini menghasilkan F1-score sebesar 66,78%. Mohan & Nair (2019) melakukan penelitian *coreference resolution* untuk kata ganti yang ambigu pada bahasa Inggris menggunakan model BERT dan SVM. Pertama, dataset dilatih pada model BERT yang didalamnya terdapat kontekstual *embedding* kemudian mengaplikasikannya pada metode *Support Vector Machine* (SVM) untuk pengklasifikasian. Penelitian ini menggunakan *dataset Gendered Ambiguous Pronouns* (GAP) yang dirilis oleh Google AI Language. Penelitian ini mendapatkan hasil akhir dengan nilai akurasi keseluruhan sebesar 78,35% dan nilai *f1-score* sebesar 71,50%.

Penelitian *coreference resolution* pada bahasa Indonesia telah dilakukan oleh Suherik & Purwarianti (2017). Peneliti menggunakan beberapa fitur sintaksis sederhana serta fitur leksikal untuk mendeteksi *coreference* pada kata ganti di kalimat langsung. Peneliti menggunakan metode C45 untuk melatih model dan menggunakan tiga metode berbeda untuk pengujian. Peneliti mengungkapkan bahwa kelemahan dari penelitian ini adalah rendahnya akurasi untuk kata yang sebenarnya berhubungan namun bukan sebuah *coreference*. Sistem masih menganggap “Budi”, “rumah Budi”, dan “teman Budi” merupakan satu entitas yang sama.

Kemudian Linggar Sari (2017) melakukan penelitian *coreference resolution* pada novel berbahasa Indonesia dengan kasus kata ganti tunggal menggunakan metode *Support Vector Machine*. Rata-rata akurasi yang diperoleh adalah 50,14%. Kekurangan dari penelitian ini adalah belum ada fitur untuk mendeteksi alias dari nama seseorang, misalnya “Sin” untuk alias “Sinta”. Untuk kasus kata ganti jamak penelitian pernah dilakukan oleh Muhammad Husni & Purnamasari (2018) menggunakan *Support Vector Machine* dengan memperoleh nilai akurasi sebesar 60,41%. Kekurangan dari kedua penelitian ini adalah belum mendukung pendekripsi kata ganti berimbuhan awalan seperti ‘ku-’ serta kata ganti berimbuhan akhiran ‘-ku’, ‘-mu’, ‘-nya’ dikarenakan proses *stemming* yang menghilangkan imbuhan serta belum ada penerapan *Named Entity Recognition* untuk mendeteksi entitas orang.

Selain itu, Auliarachman & Purwarianti (2020) melakukan penelitian menggunakan dataset *Manually Tagged Indonesian Corpus* yang terdiri dari 1000 kalimat berita . Peneliti menggunakan metode *mention-pair* yang menggunakan *deep neural network* untuk mempelajari hubungan antar dua entitas. Peneliti juga menggunakan *lexical* dan *syntactic feature* yang digunakan pada penelitian Suherik & Purwarianti (2017). Penelitian ini menghasilkan nilai akurasi sebesar 67,37 % tanpa *singleton classifier*, 63,27 % dengan *singleton classifier*, dan 75,95 % dengan *gold singleton classifier* pada CoNLL.

Pada penelitian ini, penulis menggunakan algoritma *Random Forest Classifier* untuk tugas *coreference resolution*. *Random Forest Classifier* adalah varian dari *Random Forest* yang khusus digunakan untuk tugas klasifikasi, yaitu memprediksi kelas atau label dari sampel. *Random Forest Classifier* merupakan algoritma pembelajaran mesin *ensemble* yang terdiri dari banyak pohon keputusan yang dilatih pada *subset* data secara acak serta menggunakan teknik *bootstrap aggregating* (*bagging*) untuk meningkatkan akurasi dan stabilitas prediksi dengan mengurangi variansi dan mencegah *overfitting* (Breiman, 2001). Algoritma ini mampu menangani data dengan banyak fitur tanpa *overfitting*, yang sangat penting dalam tugas *coreference* yang melibatkan banyak atribut (Heeyoung et al., 2017). Selain itu, *Random Forest* efektif dalam menangani data tidak seimbang, sering ditemui dalam *coreference* karena tidak semua pasangan entitas memiliki hubungan *coreference* (Rahman & Ng, 2011).

Random Forest telah terbukti dalam berbagai tugas NLP, termasuk *coreference resolution* dengan kinerja yang kompetitif dibandingkan metode lain seperti SVM dan *Decision Tree* (Mondal, 2020).

Berdasarkan latar belakang permasalahan yang telah dipaparkan, penulis akan melakukan penelitian dengan judul “*COREFERENCE RESOLUTION UNTUK TEKS BAHASA INDONESIA MENGGUNAKAN RANDOM FOREST CLASSIFIER*”.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang diuraikan, minimnya fitur ekstraksi, belum adanya pemanfaatan *Part-of-Speech Tagging* (pelabelan kelas kata) dan *Named Entity Recognition* (pengenalan entitas bernama), dan belum adanya proses pendekripsi imbuhan ‘ku-’, ‘-ku’, ‘-mu’, ‘-nya’ sebagai kata ganti kepemilikan pada penelitian sebelumnya dapat menghambat keefektifan dan keakuratan metode dan model *coreference resolution* untuk novel berbahasa Indonesia yang kemudian menghambat pemahaman teks dan analisis bahasa dalam konteks sastra.

1.3. Batasan Masalah

Agar pembahasan tidak terlalu rinci, penelitian ini mempunyai keterbatasan sebagai berikut.

1. Data yang digunakan merupakan teks novel berbahasa Indonesia dalam format .txt.
2. Kata yang mengandung imbuhan awalan ‘ku-’ dan imbuhan akhiran ‘-ku’, ‘-mu’, dan ‘-nya’ yang dideteksi sebagai kata ganti kepemilikan merupakan kata yang kelas katanya adalah kata benda.
3. Hasil keluaran sistem berupa pasangan kata hasil identifikasi yang berlabel *coreference* dan bukan *coreference*.

1.4. Tujuan Penelitian

Penelitian ini bertujuan mengembangkan metode dan model *coreference resolution* yang efektif dan akurat khusus untuk teks berbahasa Indonesia. Dengan mempertimbangkan tantangan dan kekurangan-kekurangan pada penelitian sebelumnya dengan menggunakan metode *Random Forest Classifier*.

1.5. Manfaat Penelitian

Hasil dari penelitian ini akan memberikan manfaat dalam bidang pengolahan bahasa alami dan analisis teks. Peningkatan kinerja *coreference resolution* pada novel berbahasa Indonesia akan membuka pintu bagi pengembangan aplikasi NLP yang lebih canggih dalam pemahaman teks dan analisis sastra.

1.6. Metodologi Penelitian

Metodologi yang digunakan pada penelitian ini adalah.

1.6.1. Studi literatur

Pada tahapan ini peneliti melakukan pencarian informasi yang diperolah dari buku, jurnal, skripsi serta beberapa sumber informasi lainnya. Informasi yang dihimpun berkaitan dengan novel, *Natural Language Processing* (pengolahan bahasa alami), *pre-processing* (pra-pengolahan), *Part-of-Speech Tagging* (pelabelan kelas kata), *Named Entity Recognition* (NER), *coreference resolution*, serta metode *Random Forest Classifier*.

1.6.2. Analisis permasalahan

Pada tahapan ini dilakukan analisis permasalahan menggunakan informasi yang didapatkan pada tahapan studi literatur. Hasil analisis tersebut dapat digunakan untuk menemukan metode yang tepat untuk menyelesaikan permasalahan pada penelitian ini.

1.6.3. Perancangan sistem

Pada tahapan ini analisis permasalahan penelitian, rancangan struktur program dan tampilan, analisis kebutuhan perangkat lunak serta penerapan metode *Random Forest Classifier* untuk *coreference resolution* akan ditelaah secara mendalam.

1.6.4. Implementasi dan pengujian sistem

Pada tahapan ini akan dilakukan pengimplementasian sistem gambaran tampilan aplikasi yang telah dibuat sebelumnya dan pengujian aplikasi.

1.6.5. Dokumentasi dan penyusunan laporan

Pada tahapan ini akan dilakukan dokumentasi dan penyusunan laporan dari hasil evaluasi berdasarkan penelitian yang telah dilakukan.

1.7. Sistematika Penulisan

Penelitian ini menggunakan sistematika penulisan yang terdiri atas lima bagian yang akan diuraikan sebagai berikut.

Bab 1: Pendahuluan

Bab ini berisi uraian tentang latar belakang dari penelitian, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

Bab 2: Landasan Teori

Bab ini berisi uraian teori-teori yang menjadi landasan dan penunjang untuk memahami permasalahan pada penelitian ini. Pada bab ini novel, *coreference resolution*, *Natural Language Processing* (NLP), *pre-processing* (pra-pengolahan), *Part-of-Speech Tagging* (pelabelan kelas kata), *Named Entity Recognition* (NER), metode *Random Forest Classifier* akan diuraikan secara terperinci. Di akhir bab akan ada uraian penelitian-penelitian terdahulu yang menjadi landasan rujukan.

Bab 3: Analisis dan Perancangan Sistem

Bab ini berisi uraian analisis dan perancangan sistem menggunakan metode *Random Forest Classifier* untuk tugas *coreference resolution* pada novel berbahasa Indonesia. Penjelasan mendalam mengenai arsitektur umum dari sistem yang dibangun akan diuraikan juga pada bab ini.

Bab 4: Implementasi dan Pengujian Sistem

Bab ini berisi uraian mengenai implementasi dari rancangan yang telah dipaparkan sebelumnya di bab tiga. Dipaparkan pula hasil akhir dan evaluasi pada sistem yang dibangun.

Bab 5: Kesimpulan dan saran

Pada bab terakhir ini akan dipaparkan kesimpulan dari keseluruhan proses penelitian yang telah diuraikan pada bab-bab sebelumnya serta saran yang diajukan oleh penulis yang berguna untuk pengembangan ke depan.

BAB 2

LANDASAN TEORI

2.1. Novel

Menurut Kamus Besar Bahasa Indonesia (2016), novel merupakan karangan prosa yang panjang mengandung rangkaian cerita kehidupan seseorang di sekelilingnya dengan menonjolkan watak dan sifat setiap pelaku. Novel biasanya mengandung beberapa bab atau bagian cerita yang lebih besar. Karya novel umumnya memiliki plot yang kompleks, menggambarkan kehidupan dan pengalaman tokoh-tokohnya, serta menyampaikan pesan atau tema tertentu kepada pembaca.

Ciri-ciri umum novel meliputi:

- a. Panjang: Novel umumnya lebih panjang daripada cerita pendek atau novella. Mereka dapat mencakup ratusan halaman atau lebih tergantung pada kompleksitas cerita.
- b. Cerita yang kompleks: Novel sering kali memiliki plot yang kompleks dengan konflik dan perkembangan tokoh yang mendalam.
- c. Pengembangan tokoh: Novel memberikan ruang yang lebih luas lagi bagi pengembangan tokoh daripada bentuk sastra yang lebih pendek. Tokoh-tokoh dalam novel biasanya banyak dan memiliki latar belakang kepribadian, dan perubahan yang berbeda-beda.
- d. Gaya bahasa: Novel cenderung menampilkan gaya bahasa yang bervariasi dan menggunakan deskripsi yang lebih detail dalam menggambarkan setting, tokoh, dan suasana.
- e. Penekanan pada tema: Novel seringkali menyampaikan pesan atau tema tertentu kepada pembaca. Tema-tema dalam novel dapat berkisar dari cinta, perjuangan hidup, hubungan pribadi, atau isu-isu sosial politik.

2.2. *Natural Language Processing (NLP)*

Natural Language Processing atau yang dalam bahasa Indonesia disebut dengan pengolahan bahasa alami merupakan kombinasi dari bidang ilmu komputer dan kecerdasan buatan yang berkaitan dengan pemahaman, interpretasi, dan generasi bahasa manusia oleh komputer. NLP membantu komputer mempelajari dan memahami bahasa manusia, memungkinkan komputer dapat berkomunikasi dan memahami maknanya (Prasetyo et al. 2021). Kompleksitas bahasa manusia membuat komputer sulit memahaminya.

2.3. *Pre-processing (Pra-pengolahan)*

Pre-processing (pra-pengolahan) adalah tahapan dalam pengolahan bahasa alami (NLP) yang dilakukan sebelum menerapkan metode atau model NLP pada teks. *Pre-processing* bertujuan untuk mengubah dan membersihkan teks mentah menjadi bentuk yang lebih terstruktur, terstandarisasi, dan siap untuk diolah lebih lanjut (Lourdusamy & Abraham, 2018). *Pre-processing* yang akan diterapkan pada penelitian ini adalah *filtering* (penyaringan), *sentence tokenization* (tokenisasi kalimat), *word tokenization* (tokenisasi kata), pemisahan kata berimbuhan, dan ekstraksi fitur.

2.3.1. *Filtering*

Filtering atau penyaringan merupakan adalah proses penghapusan atau penyaringan elemen yang tidak diinginkan dari teks yang sedang diproses (Lourdusamy & Abraham, 2018). Tujuan dari *filtering* dalam NLP adalah untuk mengurangi kebisingan (*noise*) atau mempertajam fokus analisis pada informasi yang relevan. Teknik filtering yang paling umum digunakan adalah *stopword removal* dan *punctuation removal* (menghapus tanda baca). *Punctuation removal* dilakukan bergantung pada tugas yang akan dilakukan selanjutnya. Jika salah satu atau dua tanda baca dibutuhkan untuk melakukan tugas NLP selanjutnya maka tanda baca-tanda baca tersebut tidak perlu dihapus.

2.3.2. Sentence tokenization

Sentence tokenization atau tokenisasi kalimat adalah proses memecah teks berbentuk paragraf menjadi kalimat-kalimat terpisah. Tujuan dari tokenisasi kalimat adalah untuk mengidentifikasi batasan-batasan kalimat dalam teks. Ini penting karena kebanyakan metode NLP bekerja dengan tingkat kalimat, dimana analisis dan pemrosesan dilakukan pada kalimat terpisah. Biasanya, tanda baca seperti tanda titik (.) digunakan sebagai petunjuk untuk memisahkan kalimat (*delimiter*).

2.3.3. Word tokenization

Word tokenization atau tokenisasi kata adalah proses lanjutan dari tokenisasi kalimat dimana dilakukan pemotongan lagi pada penggalan kalimat sebelumnya menjadi kumpulan-kumpulan kata atau token. Biasanya spasi digunakan sebagai *delimiter* dalam tokenisasi kata.

2.3.4. Pemisahan kata berimbuhan

Pemisahan kata berimbuhan dalam NLP merujuk pada proses membagi kata-kata dalam teks menjadi unit-unit terpisah mewakili akar kata (*root form*) dan imbuhan yang mendampinginya. Tujuan dari pemisahan kata berimbuhan adalah untuk mempermudah analisis morfologis dan pemahaman struktur dalam teks.

Pemisahan kata berimbuhan penting dalam beberapa tugas NLP, seperti pemrosesan bahasa Indonesia yang kaya akan prefiks (imbuhan awalan), konfiks (imbuhan awalan dan akhiran) dan sufiks (imbuhan akhiran). Dalam bahasa Indonesia, kata-kata sering kali memiliki imbuhan awalan dan akhiran yang mengubah makna dan fungsi kata tersebut.

2.3.5. Ekstraksi fitur

Ekstraksi fitur dalam NLP adalah proses mengidentifikasi dan mengekstrak informasi yang relevan atau bermanfaat dari teks atau dokumen dengan tujuan membangun representasi numerik yang dapat digunakan dalam pemodelan atau analisis lanjutan (Eklund, 2018) .

Fitur-fitur dalam NLP dapat mencakup berbagai aspek teks, termasuk sintaksis, leksikal, semantik, dan kontekstual. Beberapa contoh ekstraksi fitur dalam NLP meliputi:

1. Fitur Sintaksis

Ekstraksi fitur sintaksis melibatkan analisis struktur sintaksis dari teks, seperti dependensi antara kata-kata, tipe frase, atau posisi kata dalam kalimat.

2. Fitur Leksikal

Ekstraksi fitur leksikal berfokus pada karakteristik kata-kata kunci, kata-kata bersamaan (*collocations*), frekuensi kata, atau kelas kata (*part of speech*).

3. Fitur Semantik

Ekstraksi fitur semantik berusaha memahami makna dan hubungan antara kata-kata dalam teks. Misalnya pemetaan kata ke dalam ruang vektor semantik atau penggunaan *word embeddings* seperti *Word2Vec* atau *GloVe*.

4. Fitur Kontekstual

Ekstraksi fitur kontekstual mencakup informasi yang terkait dengan konteks atau lingkungan di mana teks ditempatkan, seperti entitas bernaam, tanggal, lokasi, atau topik tertentu.

Fitur-fitur yang berhasil diekstraksi dapat digunakan sebagai masukan untuk berbagai tugas NLP, seperti klasifikasi teks, analisis sentimen, pemahaman bahasa alami, dan lain sebagainya.

2.4. *Part-of-Speech Tagging (POS Tagging)*

Part-of-Speech Tagging (POS Tagging) dalam NLP merupakan teknik pelabelan atau penandaan jenis kelas kata (*part-of-speech*) terhadap setiap kata dalam teks (Fanoon & Uwanthika, 2019). Jenis kelas kata tersebut mencakup kata kerja, kata benda, kata sifat, kata ganti, kata keterangan, konjungsi, dan sebagainya. Tujuan POS *tagging* adalah untuk mengidentifikasi peran dan peran kata dalam konteks kalimat agar lebih memahami lebih sintaksis, gramatikal, dan penggunaan kata dalam suatu teks. Dinakaramani *et al.* (2014) telah merancang daftar *tagset* yang dapat diidentifikasi oleh POS *tagger*. Daftar *tagset* ditunjukkan pada Tabel 2.1.

Tabel 2.1 Daftar Tagset

No.	Tag	Deskripsi	Contoh
1	CC	Konjungtor koordinatif Numeralia kardinal.	atau, tetapi, dan satu, ratus, sepuluh, 7916, sepertiga, 0,010, 0,50, puluhan, 2008, 24
2	CD		
3	OD	Numeralia ordinal	kelima, kesepuluh, ke-5
4	DT	Artikel	sang, si, para
5	FW	Kata bahasa asing	<i>before, after, weather</i>
6	IN	Preposisi atau kata depan	dalam, pada, untuk, ke, di
7	JJ	Adjektiva atau kata sifat	kotor, pendek, putih, cepat, senang, bahagia, kotak
8	MD	Verba modal dan verba bantu	perlu, mesti, harus
9	NEG	Kata ingkar	jangan, tidak, belum
10	NN	Nomina atau kata benda	rumah, pensil, pesawat, komputer
11	NNP	<i>Proper noun</i> atau kata benda sepsifik	Susi, Laut Hitam, Piala Dunia, Liga Spanyol, Idul Adha
12	NND	Penggolong atau nomina ukuran	lembar, helai, ton, orang
13	PR	Demonstrative atau pronomina petunjuk	situ, sini, ini, itu
14	PRP	Pronomina persona	dia, saya, kamu, kita, mereka, kalian
15	RB	Adverbia atau kata keterangan	segera, niscaya, justru, sangat
16	RP	Partikel	-kah, -lah, pun
17	SC	Konjungtor subordinatif	semoga, jika, supaya, maka, sebab, bahwa, sejak
18	SYM	Simbol	@, #, \$, %, &
19	UH	Interjeksi	hai, ayo, duh, oh, brengsek
20	VB	Verba atau kata kerja	makan, meliha, bekerja, menyapu
21	WH	Pronomina penanya atau kata tanya	apa, kemana, mengapa, bagaimana, siapa
22	x	Tidak diketahui	statemen
23	Z	Tanda baca	“...”,?..

Dalam praktiknya, *POS Tagging* dapat dilakukan menggunakan berbagai metode, termasuk berbasis aturan, kamus kata, atau pendekatan berbasis pembelajaran mesin seperti model statistik (model *Hiden Markov* atau HMM (Widhiyanti & Harjoko, 2012)) dan model *Conditional Random Fields* atau CRF (Fanoon & Uwanthika, 2019), atau model berbasis jaringan saraf (LSTM atau Transformer (Alkhwiter & Al-Twairesh, 2020)).

2.5. *Named Entity Recognition (NER)*

Named Entity Recognition (NER) atau pengenalan entitas bernaama digunakan untuk mendeteksi informasi dalam teks dan mengklasifikasikannya kedalam sebuah set kategori (Fanny, et al. 2022). Set kategori tersebut biasanya berupa entitas bernaama seperti nama orang, lokasi, organisasi, dan waktu.

Terdapat dua pendekatan utama untuk NER, *rule-based*, dan *machine-learning based* (Fanny, et al. 2022). Beberapa pendekatan untuk NER bergantung pada *POS Tagging*. NER umumnya dilihat sebagai masalah prediksi sekuensial yang bertujuan untuk menetapkan label yang benar pada masing-masing token. Berbagai macam sistem pengkodean yang digunakan dalam NER. Dua pendekatan yang populer adalah BIO dan BILOU (Amaral et al. 2015).

Pendekatan BIO digunakan untuk mengklasifikasikan bagian-bagian dari teks ke dalam tiga kategori utama, yaitu: B (*Begin*), I (*Inside*), dan O (*Outside*).

1. B (*Begin*) mewakili awal dari sebuah entitas. Misalnya, “B-PERSON” digunakan untuk menandai awal entitas orang.
2. I (*Inside*) menandakan bagian tengah dari entitas yang sama. Ini digunakan untuk kata-kata yang mengikuti kata awal (B) dari entitas yang sama. Misalnya, “I-PERSON” akan digunakan untuk kata-kata yang merupakan bagian dari nama orang yang sama.
3. O (*Outside*) menandakan kata yang tidak termasuk dalam kategori entitas apapun. Kata-kata yang tidak relevan dengan entitas yang diidentifikasi akan diberi label “O”.

Berikut contoh penggunaan skema BIO dalam NER pada Gambar 2.1.

Joe	→	B-PERSON
Biden	→	I-PERSON
adalah	→	O
Presiden	→	O
Amerika	→	B-LOCATION
Serikat	→	I-LOCATION

Gambar 2.1 Pendekatan BIO pada NER

Pendekatan BILOU digunakan untuk mengklasifikasikan bagian-bagian dari teks ke dalam lima kategori utama, yaitu sebagai berikut.

1. B (*Begin*), mewakili awal dari sebuah entitas. Misalnya, “B-PERSON” digunakan untuk menandai awal entitas orang.
2. I (*Inside*), menandakan bagian tengah dari entitas yang sama. Ini digunakan untuk kata-kata yang mengikuti kata awal (B) dari entitas yang sama. Misalnya, “I-PERSON” akan digunakan untuk kata-kata yang merupakan bagian dari nama orang yang sama.
3. L (*Last*), menandakan kata terakhir dalam suatu entitas. Misalnya, “L-PERSON” akan digunakan untuk menandai kata terakhir dalam entitas orang.
4. U (*Unit*), digunakan ketika suatu entitas hanya terdiri dari satu suku kata. Misalnya, “U-LOCATION” akan digunakan untuk menandai kata yang merupakan entitas lokasi tunggal.
5. O (*Outside*), menandakan kata yang tidak termasuk dalam kategori entitas apapun. Kata-kata yang tidak relevan dengan entitas yang diidentifikasi akan diberi label “O”.

Berikut contoh penggunaan skema BILOU dalam NER pada Gambar 2.2.

Joko	→	B-PERSON
Widodo	→	L-PERSON
adalah	→	O
Presiden	→	O
Indonesia	→	U-LOCATION

Gambar 2.2 Pendekatan BILOU pada NER

2.6. *Coreference Resolution*

Coreference Resolution merupakan subtugas *Natural Language Processing* (NLP) yang digunakan untuk menentukan apakah satu atau lebih kata dalam teks merujuk ke entitas yang sama dalam teks. Entitas tersebut dapat berupa orang, tempat, objek, atau bahkan konsep abstrak. Ilustrasi bagaimana *coreference resolution* bekerja dapat dilihat pada Gambar 2.3.

Bagi **Sherlock Holmes₁**, **dia₂** adalah wanita yang istimewa. **Dia₃** tak pernah menyebut wanita itu dengan istilah lain. Di matanya wanita itulah yang paling hebat di antara seluruh kaumnya₄. Ini tidak berarti bahwa **Holmes₅** mencintai **Irene Adler₆**.

Gambar 2.3 Ilustrasi *coreference*

Pada teks diatas, terdapat 2 buah entitas orang yaitu “Sherlock Holmes₁” dan “Irene Adler₆”. *Coreference resolution* akan mengidentifikasi bahwa “dia₂” dan imbuhan “nya₄” yang merupakan kata ganti kepemilikan merujuk kepada “Irene Adler₆” serta “Dia₃” dan “Holmes₅” merujuk kepada “Sherlock Holmes₁”.

Memahami hubungan antara frasa atau kata yang merujuk kepada entitas yang sama dapat membantu sistem NLP dalam memahami konteks dan menjalankan tugas lain seperti analisis sentimen, ekstraksi informasi, dan pemahaman teks yang lebih mendalam.

Terdapat beberapa teknik dalam melakukan *coreference resolution*, yaitu:

1. *Mention-Pair* model: Menggunakan model statistik untuk membandingkan pasangan frasa yang mungkin merujuk kepada entitas yang sama.
2. *End-to-End Neural* model: Menggunakan jaringan saraf tiruan untuk memodelkan hubungan *coreference* secara langsung.
3. *Clustering algorithms*: Menggunakan algoritma klasterisasi untuk mengelompokkan entitas yang merujuk kepada hal yang sama.

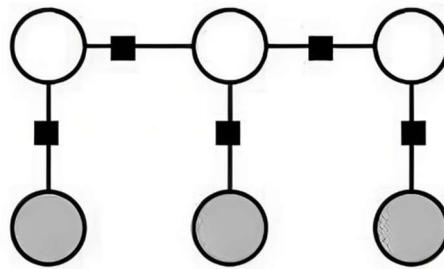
Dalam prosesnya, *coreference resolution* seringkali berbenturan pada hal-hal seperti keambiguitasan dimana teks sering kali mengandung frasa yang ambigu, yang dapat merujuk kepada beberapa entitas yang berbeda. Kemudian kondisi tidak eksplisit dimana entitas yang sama dalam teks mungkin tidak selalu dinyatakan secara eksplisit, sehingga perlu pemahaman konteks untuk mengidentifikasinya.

2.7. Conditional Random Fields (CRF)

Conditional Random Fields (CRF) adalah kerangka kerja untuk membangun model probabilistik diskriminatif untuk segmentasi dan pelabelan data sekuensial (Pisceldo et al. 2009) dimana CRF juga termasuk varian dari *Markov Random Field* (MRF) yang merupakan jenis model grafis tak berarah.

CRF biasanya digunakan untuk tugas prediksi terstruktur, dimana targetnya adalah memprediksi keluaran terstruktur berdasarkan satu set fitur masukan. Misalnya, pada NLP, tugas prediksi yang umumnya terstruktur adalah POS *tagging* (Fanoon & Uwanthika, 2019), dimana tujuannya adalah untuk menetapkan kelas kata ke setiap kata pada kalimat. CRF juga dapat diterapkan untuk *Named Entity Recognition* (NER), *chunking* (pemotongan), dan tugas-tugas lain yang hasil keluarannya berupa urutan terstruktur. CRF dilatih menggunakan estimasi kemungkinan maksimum, yang melibatkan pengoptimalan parameter-model untuk memaksimalkan probabilitas urutan keluaran yang benar berdasarkan fitur masukan. Masalah optimisasi ini biasanya diselesaikan menggunakan algoritma iteratif seperti *gradient descent* atau L-BFGS (Susmitha & Haritha, 2020).

CRF mempunyai beberapa pola, salah satunya adalah *linear-chain* (Gambar 2.4).



Gambar 2.4 Struktur *linear-chain* CRF (Fanny et al. 2022)

Linear-chain digunakan untuk NER karena hasil keluaran NER merupakan data sekuensial. Berikut persamaan model *linear-chain* CRF (Sokolovska et al. 2010):

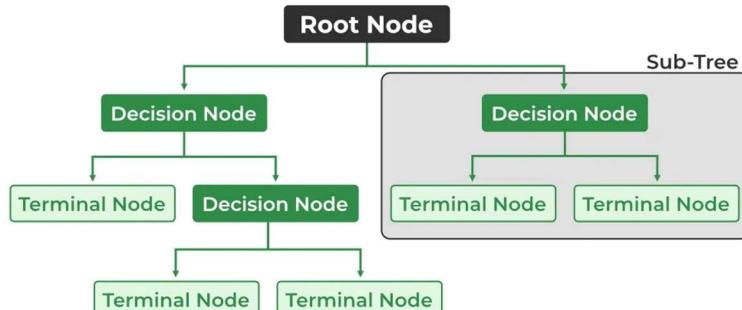
$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, x_t) \right\} \dots \dots \dots \quad (2.1)$$

Secara garis besar ada 2 komponen rumus CRF:

1. Normalisasi: Tidak ada probabilitas di sisi kanan persamaan yang memiliki bobot dan fitur. Namun, keluarannya diharapkan berupa probabilitas dan oleh karena itu perlu danya normalisasi. Konstanta normalisasi Z(X) adalah jumlah dari semua barisan keadaan yang mungkin sehingga totalnya menjadi 1.
2. Bobot dan fitur: Komponen ini dianggap sebagai rumus regresi logistik dengan bobot dan fitur yang sesuai. Estimasi bobot dilakukan dengan estimasi kemungkinan maksimum dan fiturnya.

2.8. Random Forest Classifier

Random Forest Classifier merupakan implementasi spesifik dari metode *Random Forest* yang dirancang khusus untuk tugas klasifikasi dimana metode *Random Forest* merupakan metode pembelajaran mesin *supervised learning* berdasarkan metode *ensemble learning*. *Ensemble learning* melibatkan penggabungan dari beberapa model untuk meningkatkan kinerja dan kestabilan dibandingkan dengan model tunggal. *Random Forest Classifier* menggabungkan beberapa algoritma *decision tree* untuk membuat prediksi. *Decision tree* adalah metode pembelajaran mesin non-parametrik yang berbentuk seperti struktur pohon. *Decision tree* diawali dengan satu simpul (*root node*) yang mewakili pertanyaan awal, simpul berikutnya (*decision node/internal*) yang memiliki jawaban dari pertanyaan sebelumnya, cabang yang mewakili pilihan jalur keputusan, daun (*terminal node/leaf node*) yang mewakili keputusan akhir. Berikut ilustrasi metode *decision tree* pada Gambar 2.5.

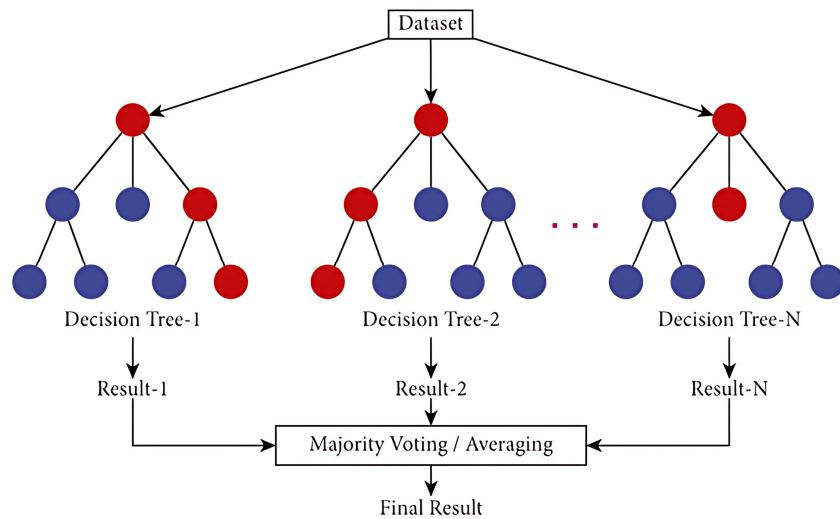


Gambar 2.5 Struktur metode *Decision Tree* (Geeksforgeeks, 2023)

Random Forest Classifier mengatasi kelemahan *decision tree* yang sering kali tidak dapat menghasilkan model yang efektif karena hanya menggunakan satu “pohon” saja. Satu *decision tree* rentan terhadap *noise* sehingga dibutuhkan metode untuk mengurangi efek *noise* tersebut dengan menggabungkan beberapa *decision tree* sehingga dapat memberikan hasil yang lebih baik. Semakin besar jumlah *decision tree* yang dibuat maka model yang dihasilkan semakin baik. Secara garis besar, metode *Random Forest* membuat sebuah hutan (*forest*) yang didalamnya terdapat sejumlah pohon (*tree*). Semakin tinggi dan semakin banyak jumlah pohon, maka semakin bagus hutan tersebut.

Random Forest Classifier juga menggunakan metode *bootstrap aggregation* (*bagging*) dimana metode ini melibatkan pembuatan *subset* acak dari dataset pelatihan

dengan penggantian (*bootstrap*) dan melatih model pada setiap *subset* ini. Masing-masing pohon dalam *Random Forest Classifier* diberi data yang berbeda, menghasilkan pohon-pohon yang beragam. Prediksi akhir didasarkan pada agregat hasil atau *voting* dari pohon-pohon tersebut (Tantyoko et al. 2023). Ilustrasi alur kerja metode *Random Forest Classifier* dapat dilihat pada Gambar 2.6.



Gambar 2.6 Alur kerja metode *Random Forest Classifier* (Bernal, 2023)

Berikut adalah penjelasan dari cara kerja metode *Random Forest Classifier*.

1. Pembentukan *subset* acak (*bootstrapped sampling*)

- Diberikan dataset pelatihan D dengan N instance, kemudian diambil sebanyak B *subset* acak dengan penggantian menggunakan proses *bootstrap sampling*. Setiap *subset* D_i memiliki n *instance* yang dipilih secara acak dari D .

2. Pembangunan pohon keputusan

- Untuk setiap D_i , pohon keputusan T_i dibangun dengan menggunakan pemilihan fitur acak pada setiap simpul. Misal dapat direpresentasikan pohon keputusan sebagai $T_i(X)$, dimana X adalah vektor fitur.
- Pada setiap simpul j dari T_i , dipilih fitur acak f_j dari *subset* fitur yang dipilih secara acak. Kriteria pemisahan, misalnya *Gini impurity* atau *entropi*, digunakan untuk memilih nilai ambang pemisahan t_j .
- Pada dasar pohon keputusan, kelas prediksi c dapat dihasilkan berdasarkan mayoritas kelas dari *instance* yang termasuk dalam daun tertentu.

- Proses ini diulangi hingga mencapai kondisi berhenti (misalnya, kedalaman maksimum tercapai atau ukuran simpul minimum terpenuhi).

3. *Random feature selection*

- Untuk setiap pohon T_i , dipilih sejumlah kecil fitur acak (m) dari M fitur total. Ini membantu meningkatkan keberagaman antar pohon.

4. Prediksi dari setiap pohon

- Setelah semua pohon (T_i) dibangun, kelas untuk *instance* baru dapat diprediksi dengan menjalankan *instance* tersebut melalui setiap pohon dan mengambil mayoritas voting dari prediksi kelas (c).

$$\hat{y}(X) = \text{mode} \{T_i(X) \text{ for } i = 1, 2, 3, \dots, B\}$$

5. Output akhir

- Prediksi akhir dari *Random Forest Classifier* adalah hasil dari mayoritas voting (klasifikasi) dari prediksi-prediksi semua pohon dalam *ensemble*.

2.9. Penelitian Terdahulu

Penelitian berkaitan dengan *coreference resolution* telah banyak diteliti dalam berbagai bahasa seperti bahasa Korea, Cina, Inggris, dan Indonesia. Penelitian pada bahasa Korea dilakukan oleh Seok-Won *et al.* (2016) menggunakan metode *Random Forest* dengan *sieve-guided features* menggunakan data berita bisbol sebanyak 272 dokumen serta dilakukan teknik *10-fold cross-validation* pada data *train*. Peneliti juga membandingkan penelitiannya dengan metode lain seperti *Support Vector Machine* (SVM) dan *Maximum Entropy Model* (MEM). Hasil terbaik didapatkan oleh metode *Random Forest* dengan F1-score sebesar 66,78%.

Mohan & Nair (2019) melakukan penelitian *coreference resolution* untuk kata ganti yang ambigu pada bahasa Inggris menggunakan model BERT dan SVM. Pertama, dataset dilatih pada model BERT yang didalamnya terdapat kontekstual embedding kemudian mengaplikasikannya pada metode *Support Vector Machine* (SVM) untuk pengklasifikasian. *Dataset* yang digunakan dalam penelitian ini adalah *Gendered Ambiguous Pronouns* (GAP) yang dirilis oleh Google AI Language. Penelitian ini mendapatkan hasil akhir dengan nilai akurasi keseluruhan sebesar 78,35% dan nilai F1-score sebesar 71,50%.

Penelitian *coreference resolution* pada bahasa Indonesia telah dilakukan oleh Suherik & Purwarianti (2017). Peneliti menggunakan beberapa fitur sintaksis sederhana serta fitur leksikal untuk mendeteksi *coreference* pada kata ganti di kalimat langsung pada 200 kalimat berita. Peneliti menggunakan metode C45 untuk melatih model dan menggunakan tiga metode berbeda untuk pengujian. Didapatkan nilai F1-score akhir sebesar 71,6%. Peneliti mengungkapkan bahwa kelemahan dari penelitian ini adalah rendahnya akurasi untuk kata yang sebenarnya berhubungan namun bukan sebuah *coreference*. Sistem masih menganggap “Budi”, “rumah Budi”, dan “teman Budi” merupakan satu entitas yang sama.

Kemudian Linggar Sari (2017) melakukan penelitian *coreference resolution* pada novel berbahasa Indonesia dengan kasus kata ganti tunggal menggunakan metode *Support Vector Machine*. Rata-rata akurasi yang diperoleh adalah 50,14%. Kekurangan dari penelitian ini adalah belum ada fitur untuk mendeteksi alias dari nama seseorang, misalnya “Sin” untuk alias “Sinta”. Untuk kasus kata ganti jamak penelitian pernah dilakukan oleh Muhammad Husni & Purnamasari (2018) menggunakan *Support Vector Machine* dengan memperoleh nilai akurasi sebesar 60,41%. Kekurangan dari kedua penelitian ini adalah belum mendukung pendekripsi kata ganti berimbuhan awalan seperti ‘ku-’ serta kata ganti berimbuhan akhiran ‘-ku’, ‘-mu’, ‘-nya’ dikarenakan proses *stemming* yang menghilangkan setiap kata ganti imbuhan serta belum ada penerapan *Named Entity Recognition* untuk mendeteksi entitas orang.

Selain itu, Auliachman & Purwarianti (2020) melakukan penelitian menggunakan dataset *Manually Tagged Indonesian Corpus* yang terdiri dari 1000 kalimat berita . Peneliti menggunakan metode *mention-pair* yang menggunakan *deep neural network* untuk mempelajari hubungan antar dua entitas. Peneliti juga menggunakan *lexical* dan *syntactic feature* yang digunakan pada penelitian Suherik & Purwarianti (2017). Penelitian ini menghasilkan nilai akurasi sebesar 67,37 % tanpa *singleton classifier*, 63,27 % dengan *singleton classifier*, dan 75,95 % dengan *gold singleton classifier* pada CoNLL. Rincian singkat penelitian terdahulu ditunjukkan pada Tabel 2.2.

Tabel 2.2 Penelitian Terdahulu

No.	Peneliti	Metode	Keterangan
1	Seok-Won <i>et al.</i> (2016)	<i>Random Forest</i>	Menggunakan metode <i>Random Forest</i> , <i>sieve-guided features</i> , dan <i>10-fold cross-validation</i> pada 272 dokumen berita berita bisbol berbahasa Korea. Dilakukan juga perbandingan antara <i>Random Forest</i> dengan <i>Support Vector Machine</i> dan <i>Maximum Entropy Model</i> . Didapatkan hasil tertinggi oleh metode <i>Random Forest</i> dengan nilai F1-score sebesar 66,78%.
2	Suherik & Purwarianti (2017)	C45	Menggunakan metode C45 dan beberapa fitur sintaksis dan leksikal pada 200 kalimat berita berbahasa Indonesia. Didapatkan nilai F1-score akhir sebesar 71,6%.
3	Lingga Sari (2017)	<i>Support Vector Machine</i> (SVM)	Menggunakan metode <i>Support Vector Machine</i> (SVM) pada teks novel bahasa Indonesia untuk kata ganti tunggal. Rata-rata akurasi yang diperoleh adalah 50,14%.
4	Muhammad Husni & Purnamasari (2018)	<i>Support Vector Machine</i> (SVM)	Menggunakan metode <i>Support Vector Machine</i> (SVM) pada teks novel bahasa Indonesia dengan kasus kata ganti jamak. Akurasi yang diperolah adalah 60,41%.

Tabel 2.2 Penelitian Terdahulu (lanjutan)

No.	Peneliti	Metode	Keterangan
5	Mohan & Nair (2019)	BERT dan <i>Support Vector Machine</i> (SVM)	Menggunakan dataset <i>Gendered Ambiguous Pronouns</i> (GAP) yang dirilis oleh Google AI Languange dan dilatih pada model BERT terlebih dahulu kemudian metode <i>Support Vector Machine</i> (SVM) untuk pengklasifikasi. Akurasi keseluruhan sebesar 78,35% dan F1-score sebesar 71,50%.
6	Auliarachman & Purwarianti (2020)	<i>Deep Neural Network</i>	Mengombinasikan metode <i>mention-pair</i> dan <i>deep neural network</i> serta <i>lexical</i> dan <i>syntactic feature</i> yang digunakan pada penelitian Suherik & Purwarianti (2017) pada 1000 kalimat berita. Penelitian ini menghasilkan nilai akurasi sebesar 67,37 % tanpa <i>singleton classifier</i> , 63,27 % dengan <i>singleton classifier</i> , dan 75,95 % dengan <i>gold singleton classifier</i> pada CoNLL.

2.10. Perbedaan Penelitian

Hal yang membedakan penelitian ini dengan semua penelitian terdahulu adalah belum ada penelitian *coreference resolution* untuk teks bahasa Indonesia yang menggunakan metode *Random Forest Classifier*. Selain itu penelitian *coreference resolution* untuk teks bahasa Indonesia belum ada yang memanfaatkan *Part-of-Speech* (POS tagging) dan *Named Entity Recognition* (NER) untuk pendeksi entitas orang.. Hal lain yang membedakan adalah pada penelitian ini dilakukan pemisahan imbuhan awalan ‘ku-’ serta imbuhan akhiran ‘-ku’, ‘-mu’, ‘-nya’ yang menjadi kata ganti kepemilikan yang dapat menjadi kata rujukan dalam *coreference resolution*.

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1. Data yang digunakan

Penelitian ini menggunakan data teks novel berbahasa Indonesia berformat .txt yang dikumpulkan secara manual dari berbagai judul novel fisik dan *electronic book (e-book)*. Untuk data *train*, teks yang diambil dari novel merupakan satu bagian cerita dalam novel tersebut yang biasanya disebut satu bab atau *chapter* sedangkan untuk data *test*, teks yang diambil merupakan potongan teks dari novel yang diambil secara acak. Jumlah judul novel untuk data *train* sebanyak 18 judul dan untuk data *test* sebanyak 10 judul. Masing-masing teks dari judul novel berbeda disimpan ke dalam file .txt berbeda pula. *Dataset* yang digunakan secara rinci ditunjukkan pada Tabel 3.1.

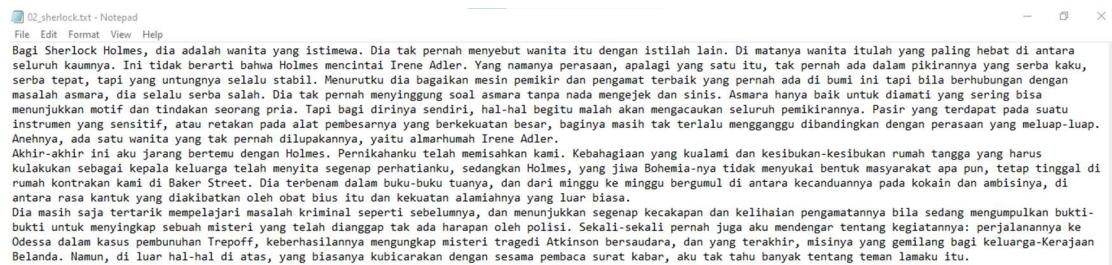
Tabel 3.1 Rincian *dataset* yang digunakan

Dataset	Judul novel	Jumlah kalimat
<i>Train</i>	Sherlock Holmes	18
	<i>Sunshine Becomes You</i>	206
	Sabtu Bersama Bapak	71
	Mimpi Sejuta Dollar	47
	<i>Summer in Seoul</i>	147
	Sang Pemimpi	24
	Hujan	101
	Hafalan Shalat Delisa	114
	Laut Bercerita	139
	Bulan	173
<i>Test</i>	Sang Alkemis	6
	Dilan	15
	Filosofi Kopi	19

Tabel 3.1 Rincian dataset yang digunakan (lanjutan)

Dataset	Judul novel	Jumlah kalimat
	Laskar Pelangi	29
	Pulang	95
	Ayat-Ayat Cinta	24
	Dalam Mihrab Cinta	29
	Ronggeng Dukuh Paruk	19
	<i>Refrain</i>	13
	86	20
	Dia Tanpa Aku	19
	Diskoneksi	20
	Jingga Untuk Matahari	15
	Seperti Dendam, Rindu	7
<i>Test</i>	Harus Dibayar Tuntas	
	Mawar Merah	26
	Maryam	28
	Petualangan Anak	21
	Natuna	
	7 Hari Menembus Waktu	29

Berikut merupakan contoh teks dari salah satu novel.



Bagi Sherlock Holmes, dia adalah wanita yang istimewa. Dia tak pernah menyebut wanita itu dengan istilah lain. Di matanya wanita itulah yang paling hebat di antara seluruh kaumnya. Ini tidak berarti bahwa Holmes mencintai Irene Adler. Yang namanya perasaan, apalagi yang satu itu, tak pernah ada dalam pikirannya yang serba kaku, serba tepat, tapi yang untungnya selalu stabil. Menurutku dia bagaikan mesin penikir dan pengamat terbaik yang pernah ada di bumi ini tapi bila berhubungan dengan masalah asmara dia selalu serba salah. Dia tak pernah menyinyalung soal asmara tanpa noda mengejek dan sinis. Asmara hanya baik untuk diamati yang sering bisa menunjukkan motif dan tindakan seorang pria. Tapi bagi dirinya sendiri, hal-hal begitu malah akan mengacaukan seluruh pemikirannya. Pasir yang terdapat pada suatu instrumen yang sensitif, atau retakan pada alat pembesaranya yang berkekuatan besar, baginya masih tak terlalu mengganggu dibandingkan dengan perasaan yang meluap-luap. Anehnya, ada satu wanita yang tak pernah dilupakannya, yaitu almarhumah Irene Adler.

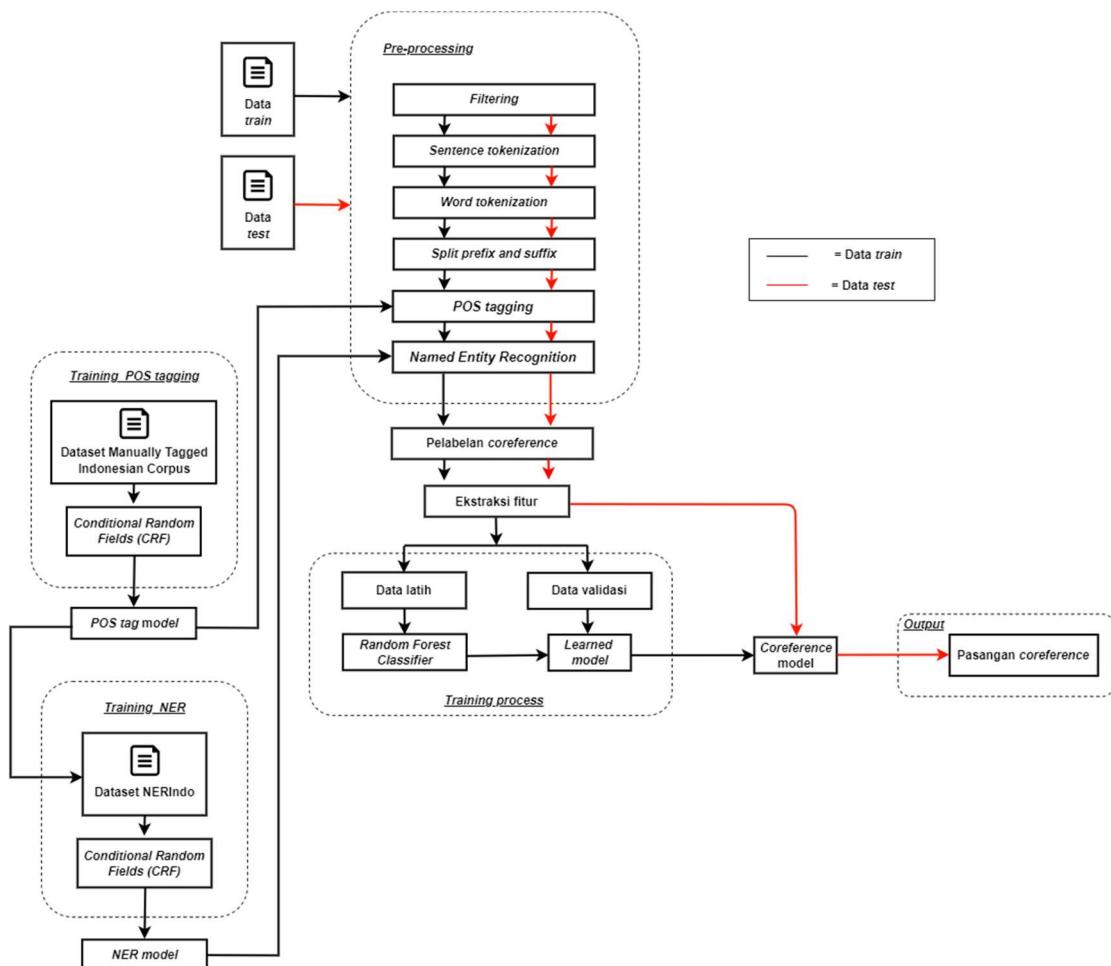
Akhir-akhir ini aku jarang bertemu dengan Holmes. Pernikahanku telah memisahkan kami. Kebahagian yang kualami dan kesibukan-kesibukan rumah tangga yang harus kulakukan sebagai kepala keluarga telah menyita perhatianku, sedangkan Holmes, yang jiwa Bohemia-nya tidak menyukai bentuk masyarakat apa pun, tetap tinggal di rumah kontrakan kami di Baker Street. Dia terkenan dalam buku-buku tuanya, dan dari minggu ke minggu bergumul di antara kecanduanya pada kokain dan ambisinya, di antara rasa kantuk yang diakibatkan oleh obat bius itu dan kekuatan alamiahnya yang luar biasa.

Dia masih saja tertarik mempelajari masalah kriminal seperti sebelumnya, dan menunjukkan segenap kecakapan dan kelihiana pengamatannya bila sedang mengumpulkan bukti-bukti untuk menyingkap sebuah misteri yang telah dianggap tak ada harapan oleh polisi. Sekali-sekali pernah juga aku mendengar tentang kegiatannya: perjalanan ke Odessa dalam kasus pembunuhan Treppoff, keberhasilannya mengungkap misteri tragedi Atkinson bersaudara, dan yang terakhir, misinya yang gemilang bagi keluarga-Kerajaan Belanda. Namun, di luar hal-hal di atas, yang biasanya kubicarakan dengan sesama pembaca surat kabar, aku tak tahu banyak tentang teman lamaku itu.

Gambar 3.1 Teks novel Sherlock Holmes

3.2. Arsitektur Umum

Metode yang diterapkan dalam penelitian ini memiliki beberapa tahapan. Tahap pertama adalah pengumpulan data *train* dan data *test* yang berupa teks novel dari beberapa novel yang kemudian disimpan dalam file berformat .txt. Selanjutnya data selanjutnya akan melalui tahapan *pre-processing* yang mencakup *filtering*, *sentence tokenization*, *word tokenization*, pemisahan prefiks dan sufiks, *POS tagging*, *named entity recognition*, pelabelan *coreference* serta ekstraksi fitur. Data *train* akan dipisahkan menjadi data *train* dan validasi. Selanjutnya data *train* akan dilatih menggunakan *random forest classifier* dan dilakukan evaluasi model terhadap data validasi guna menghasilkan model yang baik. Langkah terakhir adalah pengimplementasian model pada data *test* sehingga menghasilkan pasangan *coreference*. Untuk lebih jelasnya, proses akan diuraikan pada Gambar 3.2.



Gambar 3.2 Arsitektur Umum

3.3. Pre-processing (Pra-pengolahan)

Teks data *train* dan *test* dalam file *.txt* akan melalui serangkaian pengolahan guna untuk membersihkan dan mempersiapkan data teks agar dapat diinterpretasikan dengan baik oleh model. *Pre-processing* dilakukan karena data teks merupakan data tidak terstruktur dan mengandung *noise* (informasi tidak relevan) yang dapat memengaruhi kinerja model. Tahapan *pre-processing* yang akan dilakukan pada penelitian ini adalah *filtering*, *sentence tokenization*, *word tokenization*, pemisahan imbuhan (prefiks dan sufiks), *POS tagging*, *named entity recognition*, pelabelan *coreference*, *pairing*, ekstraksi fitur. Berikut penjelasan detail mengenai masing-masing tahapan.

3.3.1. Filtering

Pada tahapan *filtering* dilakukan penghapusan elemen-elemen yang dianggap *noise* atau gangguan dalam teks, seperti tanda baca, karakter khusus, dan informasi yang tidak diperlukan. Tujuan dari tahapan ini adalah untuk membersihkan teks agar lebih fokus pada informasi yang penting dan relevan. Pada penelitian ini tanda baca atau karakter yang dihilangkan adalah selain karakter ‘Aa-Zz’, tanda hubung (-), tanda koma (,), tanda titik (.), dan spasi. Diagram alur dari proses *filtering* dapat dilihat pada Gambar 3.3 dan penerapannya ditunjukkan pada Tabel 3.2.



Gambar 3.3 Diagram alur proses *filtering*

Tabel 3.2 Penerapan proses *filtering*

Sebelum proses <i>filtering</i>	Setelah proses <i>filtering</i>
Bagi Sherlock Holmes, dia adalah wanita yang istimewa. Dia tak pernah menyebut wanita itu dengan istilah lain. Di matanya wanita itulah yang paling hebat di antara seluruh kaumnya. Ini tidak berarti bahwa Holmes mencintai Irene Adler...	Bagi Sherlock Holmes, dia adalah wanita yang istimewa. Dia tak pernah menyebut wanita itu dengan istilah lain. Di matanya wanita itulah yang paling hebat di antara seluruh kaumnya. Ini tidak berarti bahwa Holmes mencintai Irene Adler.

3.3.2. *Sentence tokenization*

Sentence tokenization atau tokenisasi kalimat adalah tahapan dimana sebuah teks dipecah menjadi beberapa kalimat terpisah atau unit-token kalimat. Tujuan utama dari tokenisasi kalimat adalah untuk menguraikan teks menjadi unit yang lebih kecil yaitu kalimat-kalimat, sehingga memudahkan analisis dan pemrosesan lebih lanjut. Pemisahan kalimat dapat dilakukan berdasarkan tanda baca titik. Diagram alur dari proses *sentence tokenization* dapat dilihat pada Gambar 3.4 dan penerapannya ditunjukkan pada Tabel 3.3.

**Gambar 3.4 Diagram alur proses *sentence tokenization***

Tabel 3.3 Penerapan proses *sentence tokenization*

Sebelum proses <i>sentence tokenization</i>	Setelah proses <i>sentence tokenization</i>
No.	Kalimat
1	Bagi Sherlock Holmes, dia adalah wanita yang istimewa. Dia tak pernah menyebut wanita itu dengan istilah lain. Di matanya wanita itulah yang paling hebat di antara seluruh kaumnya. Ini tidak berarti bahwa Holmes mencintai Irene Adler.
2	Dia tak pernah menyebut wanita itu dengan istilah lain.
3	Di matanya wanita itulah yang paling hebat di antara seluruh kaumnya.
4	Ini tidak berarti bahwa Holmes mencintai Irene Adler.

3.3.3. *Word tokenization*

Word tokenization atau tokenisasi kata adalah proses lanjutan dari tokenisasi kalimat dimana menguraikan kalimat menjadi unit-unit yang lebih kecil lagi yang disebut token. Biasanya spasi (' ') digunakan sebagai *delimiter* dalam tokenisasi kata. Diagram alur dari proses *word tokenization* dan penerapannya dapat dilihat pada Gambar 3.5 dan penerapannya ditunjukkan pada Tabel 3.4.

**Gambar 3.5 Diagram alur proses *word tokenization***

Tabel 3.4 Penerapan proses *word tokenization*

No.	Sebelum proses <i>word tokenization</i>	Setelah proses <i>word tokenization</i>		
1	Bagi Sherlock Holmes, dia adalah wanita yang istimewa.	Bagi , wanita .	Sherlock dia yang .	Holmes adalah istimewa .
2	Dia tak pernah menyebut wanita itu dengan istilah lain.	Dia menyebut dengan .	tak wanita istilah .	pernah itu lain .
3	Di matanya wanita itulah yang paling hebat di antara seluruh kaumnya.	Di itulah hebat seluruh .	matanya yang paling di antara kaumnya .	wanita .
4	Ini tidak berarti bahwa Holmes mencintai Irene Adler.	Ini bahwa Irene .	tidak Holmes Adler .	berarti mencintai .

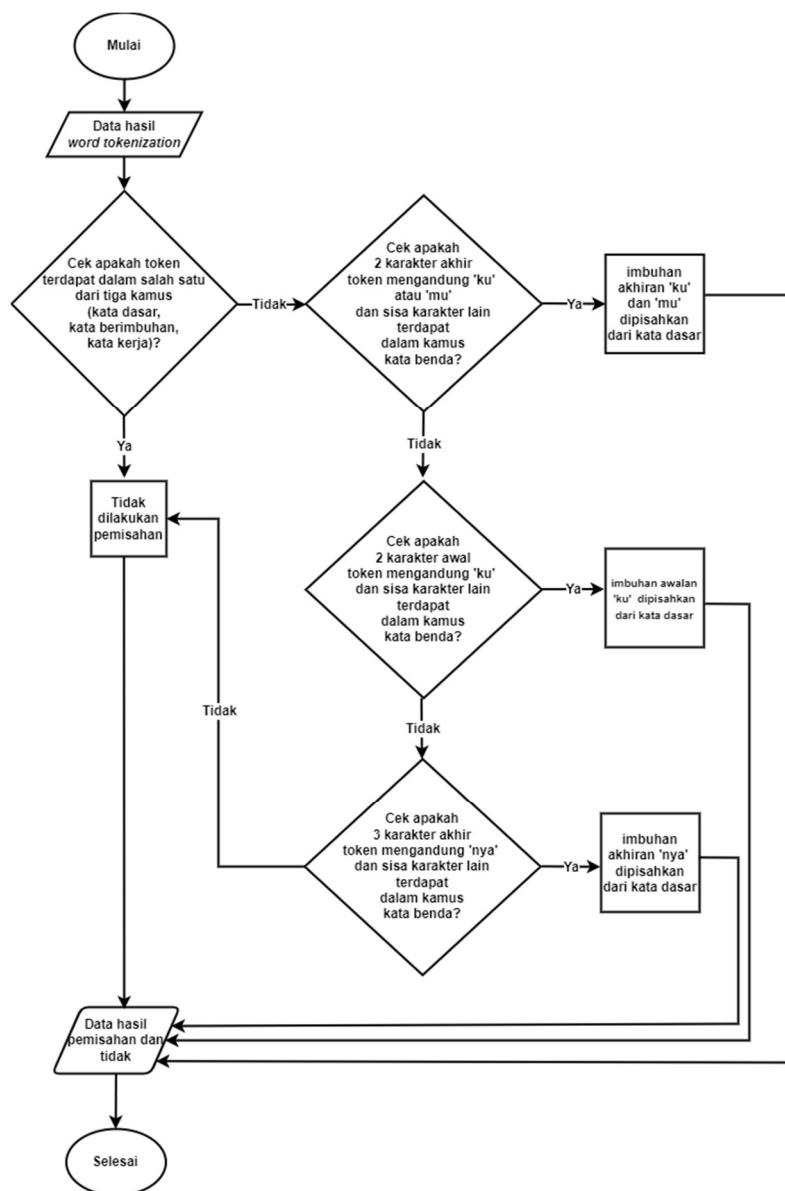
3.3.4. Pemisahan kata berimbuhan

Pada proses ini token dari hasil proses tokenisasi kata dipisahkan dari prefiks (imbuhan awalan) dan sufiks (imbuhan akhiran) yang menggantikan kata ganti kepemilikan. Sufiks dan prefiks yang dipisahkan dari token adalah sebagai berikut.

1. Prefiks ‘ku-’ sebagai kata ganti kepemilikan orang pertama tunggal.
2. Sufiks ‘-ku’ sebagai kata ganti kepemilikan orang pertama tunggal.
3. Sufiks ‘-mu’ sebagai kata ganti kepemilikan orang kedua tunggal.
4. Sufiks ‘-nya’ sebagai kata ganti orang ketiga tunggal.

Proses pemisahan dilakukan menggunakan acuan kamus kata dasar, kamus kata berimbuhan, kamus kata kerja dan kamus kata benda. Jika token terdapat dalam salah satu kamus kata dasar, kamus kata berimbuhan, dan kamus kata kerja maka tidak dilakukan proses pemisahan pada token tersebut. Jika tidak memenuhi maka pengecekan selanjutnya dilakukan menggunakan kamus kata benda. Jika dua karakter

terakhir dari token mengandung kata ‘ku’ atau ‘mu’ serta sisa karakter lainnya terdapat dalam kamus kata benda maka dilakukan pemisahan pada imbuhan ‘-ku’ dan ‘-mu’ tersebut dari kata dasarnya. Kemudian jika dua karakter awal dari token mengandung kata ‘ku’ serta sisa karakter lainnya terdapat dalam kamus kata benda maka dilakukan pemisahan pada imbuhan ‘ku-’ tersebut. Terakhir jika tiga karakter terakhir dari token mengandung kata ‘nya’ serta sisa karakter lainnya terdapat dalam kamus kata benda maka dilakukan juga pemisahan pada imbuhan ‘-nya’ tersebut dari kata dasarnya. Diagram alur dari penjelasan di atas dapat dilihat pada Gambar 3.6 dan penerapannya ditunjukkan pada Tabel 3.5.



Gambar 3.6 Diagram alur pemisahan kata berimbuhan

Tabel 3.5 Penerapan proses pemisahan kata berimbuhan

No.	Sebelum pemisahan kata berimbuhan	Sesudah pemisahan kata berimbuhan	
	Bagi	Bagi	
	Sherlock	Sherlock	
	Holmes	Holmes	
	,	,	
1	dia	dia	
	adalah	adalah	
	wanita	wanita	
	yang	yang	
	istimewa	istimewa	
	.	.	
	Dia	Dia	
	tak	tak	
	pernah	pernah	
	menyebut	menyebut	
2	wanita	wanita	
	itu	itu	
	dengan	dengan	
	istilah	istilah	
	lain	lain	
	.	.	
	Di	Di	
	matanya	mata	nya
	wanita	wanita	
3	itulah	itulah	
	yang	yang	
	paling	paling	
	hebat	hebat	
	di	di	

Tabel 3.5 Penerapan proses pemisahan kata berimbuhan (lanjutan)

No.	Sebelum pemisahan kata berimbuhan	Sesudah pemisahan kata berimbuhan
	antara	antara
	seluruh	seluruh
	kaumnya	kaum nya
	.	.
	Ini	Ini
	tidak	tidak
	berarti	berarti
	bahwa	bahwa
4	Holmes	Holmes
	mencintai	mencintai
	Irene	Irene
	Adler	Adler
	.	.

3.3.5. Part-of-speech tagging (POS tagging)

Proses POS *tagging* adalah proses pemberian label atau penandaan jenis kelas kata pada setiap kata dalam sebuah teks. Kelas kata tersebut mencakup kategori kata kata kerja, kata benda, kata keterangan, kata sifat, dan sebagainya. Tujuan dari POS *tagging* adalah untuk mengidentifikasi peran dan fungsi kata dalam konteks kalimat. Penelitian ini menggunakan kumpulan *tag* atau label dari Tabel 2.1 serta *dataset Indonesian Manually Tagged Corpus* yang dirancang oleh Dinakaramani *et al.* (2014) untuk melatih model POS *tagging*. *Indonesian Manually Tagged Corpus* merupakan *dataset* teks kalimat berbahasa Indonesia yang telah diberi label *part-of-speech*-nya oleh peneliti secara manual. *Dataset* terdiri dari 9999 kalimat yang terdiri dari 256660 token, 18249 kosa kata, dan 24 *tag* atau label. *Dataset* berformat *tab separated value* (.tsv) yang tiap barisnya terdiri dari token dan nilai POS *tag* yang dipisahkan oleh karakter *tab*. Akhir kalimat ditandai dengan baris kosong. *Dataset* tersebut memiliki struktur seperti pada Gambar 3.7.

```

<kata></t><postag><\n>
<kata></t><postag><\n>
<kata></t><postag><\n>
<kata></t><postag><\n>
    <blank_space>
<kata></t><postag><\n>
.....

```

Gambar 3.7 Format *Indonesian Manually Tagged Corpus*

Model POS *tagging* dibangun pada penelitian ini dengan menggunakan metode *Conditional Random Fields* (CRF). CRF merupakan model statistik yang baik digunakan untuk memprediksi urutan label dari data sekuensial seperti POS *tagging* (Pisceldo et al.2009). Model ini mempertimbangkan konteks dari label sebelumnya dalam memprediksi label berikutnya. Penggunaan metode CRF untuk tugas seperti POS *tagging* sudah pernah dilakukan sebelumnya oleh Fanoon *et al.* (2019) dengan performa yang baik. Diagram alur dari proses POS *tagging* dapat dilihat pada Gambar 3.8 serta penerapannya ditunjukkan pada Tabel 3.6.



Gambar 3.8 Diagram alur proses POS *tagging*

Tabel 3.6 Penerapan proses POS tagging

No.	Token	POS tag
	Bagi	IN
	Sherlock	NNP
	Holmes	NNP
	,	Z
1	dia	PRP
	adalah	VB
	wanita	NN
	yang	SC
	istimewa	JJ
	.	Z
	Dia	PRP
	tak	NEG
	pernah	NN
	menyebut	VB
2	wanita	NN
	itu	PR
	dengan	IN
	istilah	NN
	lain	JJ
	.	Z
	Di	IN
	mata	NN
	nya	PRP
	wanita	NN
3	itulah	PR
	yang	SC
	paling	RB
	hebat	JJ
	di	IN
	antara	NN

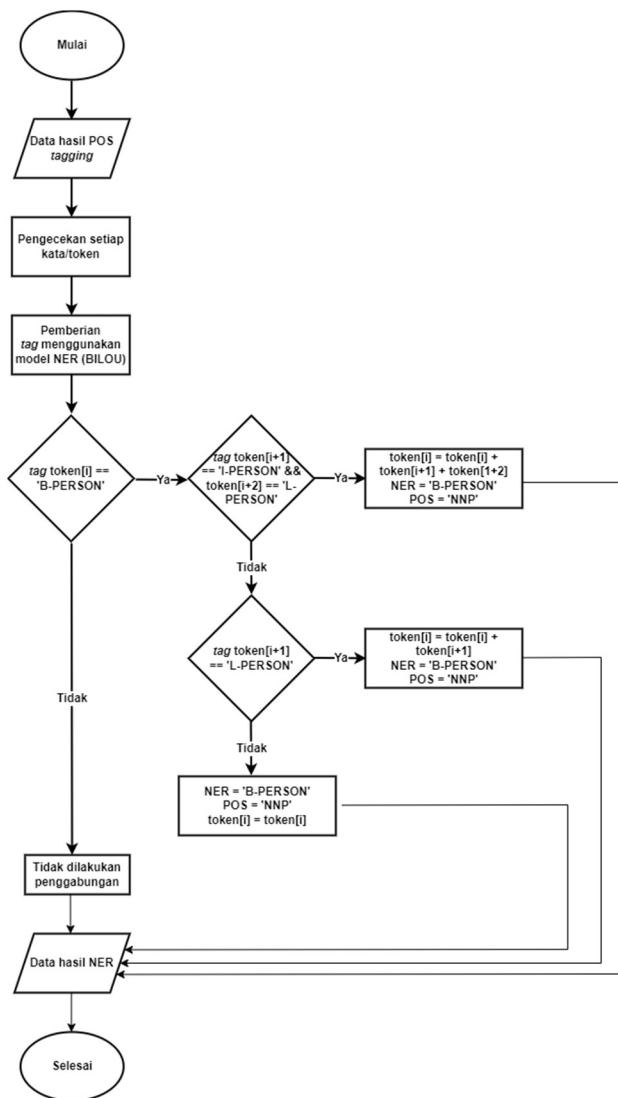
Tabel 3.6 Penerapan proses POS *tagging* (lanjutan)

No.	Token	POS tag
	seluruh	CD
	kaum	NN
	nya	PRP
	.	Z
	Ini	PR
	tidak	NEG
	berarti	VB
	bahwa	SC
4	Holmes	NNP
	mencintai	VB
	Irene	NNP
	Adler	NNP
	.	Z

3.3.6. *Named entity recognition*

Setelah proses POS *tagging* dilakukan, tahapan selanjutnya adalah mengidentifikasi entitas bernaama atau yang disebut *Named Entity Recognition* (NER). Penelitian ini menggunakan *dataset* NERIndo yang disusun oleh Syaifuddin *et al.* (2016) dengan sedikit modifikasi untuk melatih model NER. *Dataset* NERIndo berisi teks kalimat berbahasa Indonesia yang telah ditandai dengan nilai entitas pada tiap tokennya. *Dataset* terdiri dari 6288 kalimat, 118111 token, dan 14606 kosa kata. Pelabelan dalam *dataset* ini menggunakan pendekatan BILOU dimana pendekatan ini mengklasifikasikan bagian-bagian dari teks ke dalam lima kategori utama, yaitu: B (*Begin*), I (*Inside*), L (*Last*), O (*Outside*) dan U (*Unit*). *Dataset* NERIndo membagi pengklasifikasian entitas dalam lima kategori, yaitu *organization*, *location*, *time*, *person*, *quantity*, dan *o*. Masing-masing kategori memiliki empat label sesuai dengan pendekatan BILOU kecuali O. Jadi total label dalam *dataset* tersebut sebanyak 21 label. Dalam penelitian ini kategori yang digunakan hanya kategori *person* sehingga kategori selain *person* diganti menjadi label *o*. Jadi hanya lima label yang digunakan dalam penelitian ini untuk melatih model NER, yaitu *b-person*, *i-person*, *l-person*, *o*, dan *u*.

Model NER untuk penelitian ini dibangun dengan menggunakan metode yang sama dengan model POS *tagging* sebelumnya yaitu *Conditional Random Fields* (CRF). Metode CRF baik digunakan untuk memprediksi urutan label dari data sekuensial seperti POS *tagging* dan NER. Dalam pelatihan model NER digunakan pula model POS *tagging* sebelumnya sebagai salah satu fitur dalam proses pelatihan. Setelah proses NER dilakukan, selanjutkan dilakukan penggabungan entitas *person* yang memiliki label *b-person*, *i-person*, *l-person* atau *b-person*, *l-person* secara berurutan menjadi hanya satu label saja yaitu *b-person*. Penggabungan dilakukan agar nama depan, nama tengah, dan nama belakang tidak terpisah satu sama lain. Diagram alur proses NER dapat dilihat pada Gambar 3.9 serta penerapannya ditunjukkan pada Tabel 3.7.



Gambar 3.9 Diagram alur proses NER

Tabel 3.7 Penerapan proses NER

No.	Token	NER	Penggabungan	
			Token	NER
	Bagi	O	Bagi	
	Sherlock	B-PERSON	Sherlock	
	Holmes	L-PERSON	Holmes	B-PERSON
	,	O	,	O
1	dia	O	dia	O
	adalah	O	adalah	O
	wanita	O	wanita	O
	yang	O	yang	O
	istimewa	O	istimewa	O
	.	O	.	O
	Dia	O	Dia	O
	tak	O	tak	O
	pernah	O	pernah	O
	menyebut	O	menyebut	O
2	wanita	O	wanita	O
	itu	O	itu	O
	dengan	O	dengan	O
	istilah	O	istilah	O
	lain	O	lain	O
	.	O	.	O
	Di	O	Di	O
	mata	O	mata	O
	nya	O	nya	O
3	wanita	O	wanita	O
	itulah	O	itulah	O
	yang	O	yang	O
	paling	O	paling	O
	hebat	O	hebat	O

Tabel 3.7 Penerapan proses NER (lanjutan)

No.	Token	NER	Penggabungan	
			Token	NER
	di	O	di	O
	antara	O	antara	O
	seluruh	O	seluruh	O
	kaum	O	kaum	O
	nya	O	nya	O
	.	O	.	O
	Ini	O	Ini	O
	tidak	O	tidak	O
	berarti	O	berarti	O
	bahwa	O	bahwa	O
4	Holmes	U-PERSON	Holmes	U-PERSON
	mencintai	O	mencintai	O
	Irene	B-PERSON	Irene Adler	B-PERSON
	Adler	L-PERSON		
	.	O	.	O

3.4. Pelabelan *Coreference*

Pelabelan *coreference* merupakan tahapan dimana token beserta hasil dari POS *tagging*, NER, penggabungan entitas *person* sebelumnya dilabeli untuk pengklasifikasian *coreference*. Token yang dilabeli merupakan token yang memiliki kondisi seperti berikut.

1. Jika token memiliki label POS *tagging* berupa NNP dan label NER berupa *b-person* atau *u-person*.
2. Jika token memiliki label POS *tagging* berupa PRP atau token berupa kata ganti orang pertama tunggal (“ku”“saya”, “aku”), kata ganti orang kedua tunggal (“kamu”, “anda”, “kau”, “mu”, “engkau”) dan kata ganti orang ketiga tunggal (“beliau”, “dia”, “ia”, “nya”).
3. Jika token berupa kata ganti orang pertama jamak (“kami”, “kita”), kata ganti orang kedua jamak (“kalian”), atau kata ganti orang ketiga jamak (“mereka”).

Jika token memenuhi salah satu dari ketiga kondisi di atas maka akan dilakukan pelabelan *coreference* pada token tersebut. Untuk lebih jelasnya, penggambaran dari token-token yang diberi label sesuai dengan kondisi di atas ditunjukkan pada Tabel 3.8.

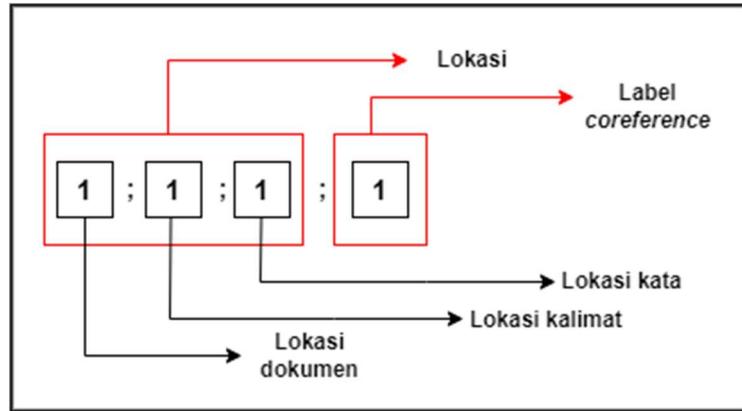
Tabel 3.8 Token-token yang diberi label

No.	Token	POS tag	NER	Label?
1	Bagi	IN	O	Tidak
2	Sherlock Holmes	NNP	B-PERSON	Ya
3	,	Z	O	Tidak
4	dia	PRP	O	Ya
5	adalah	VB	O	Tidak
6	wanita	NN	O	Tidak
7	yang	SC	O	Tidak
8	istimewa	JJ	O	Tidak
9	.	Z	O	Tidak
10	Dia	PRP	O	Ya
11	tak	NEG	O	Tidak
12	pernah	NN	O	Tidak
13	menyebut	VB	O	Tidak
14	wanita	NN	O	Tidak
15	itu	PR	O	Tidak
16	dengan	IN	O	Tidak
17	istilah	NN	O	Tidak
18	lain	JJ	O	Tidak
19	.	Z	O	Tidak
20	Di	IN	O	Tidak
21	mata	NN	O	Tidak
22	nya	PRP	O	Ya
23	wanita	NN	O	Tidak
24	itulah	PR	O	Tidak
25	yang	SC	O	Tidak

Tabel 3.8 Token-token yang diberi label (lanjutan)

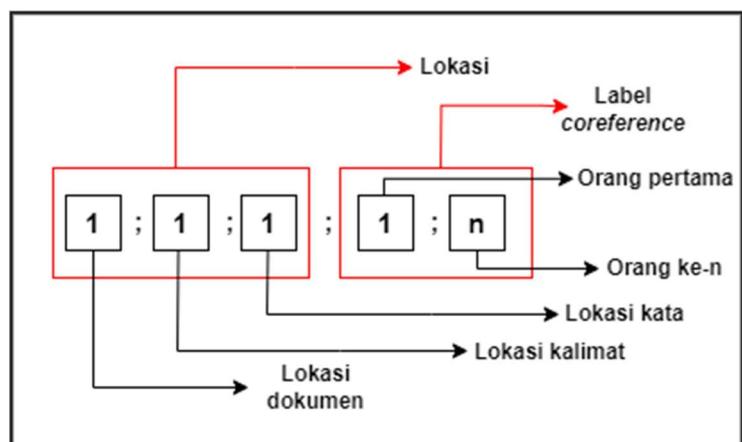
No.	Token	POS tag	NER	Label?
26	paling	RB	O	Tidak
27	hebat	JJ	O	Tidak
28	di	IN	O	Tidak
29	antara	NN	O	Tidak
30	seluruh	CD	O	Tidak
31	kaum	NN	O	Tidak
32	nya	PRP	O	Ya
33	.	Z	O	Tidak
34	Ini	PR	O	Tidak
35	tidak	NEG	O	Tidak
36	berarti	VB	O	Tidak
37	bahwa	SC	O	Tidak
38	Holmes	NNP	U-PERSON	Ya
39	mencintai	VB	O	Tidak
40	Irene Adler	NNP	B-PERSON	Ya
41	.	Z	O	Tidak

Label yang diberikan terdiri dari empat angka yang dipisahkan oleh titik koma (;) untuk token berkondisi 1 dan 2 (entitas dan kata ganti kepemilikan tunggal) dan lima angka atau lebih untuk token berkondisi 3 (kata ganti kepemilikan jamak) yang dipisahkan oleh titik koma (;) pula. Format penulisan label dapat dilihat pada Gambar 3.10 dan Gambar 3.11.



Gambar 3.10 Format pelabelan data tunggal

Format pelabelan data tunggal digunakan untuk token-token yang memenuhi kondisi 1 dan 2 yaitu berupa entitas nama dan kata ganti kepemilikan tunggal (“saya”, “aku”, “ku”, “engkau”, “kamu”, “anda”, “kau”, “mu”, “dia”, “ia”, “beliau”, “nya”). Angka pertama pada format pelabelan merupakan lokasi dokumen dimana token berada, angka kedua merupakan lokasi kalimat dimana token berada, angka ketiga merupakan lokasi kata dimana token berada, yang terakhir angka keempat merupakan label *coreference* dari token. Jika token *i* diberikan label *coreference* 1 kemudian token selanjutnya merupakan entitas atau kata ganti yang sama dengan token *i* atau merujuk pada token *i* maka token tersebut diberi label 1 juga. Token *i* dan *j* adalah *coreference* jika token *i* dan *j* memiliki label angka *coreference* yang sama. Jika token *i* diberikan label *coreference* 1 kemudian token selanjutnya bukan merupakan entitas atau kata ganti yang sama dengan token *i* atau merujuk pada token *i* maka token tersebut diberi label 2, begitu seterusnya. Token *i* dan *j* adalah bukan *coreference* jika token *i* dan *j* memiliki label angka *coreference* yang berbeda.



Gambar 3.11 Format pelabelan data jamak

Format pelabelan data jamak digunakan untuk token-token yang memenuhi kondisi 3 yaitu berupa kata ganti kepemilikan jamak (“kita”, “kami”, “kalian”, “mereka”). Teknis pelabelan data jamak sama dengan data tunggal. Yang menjadi perbedaan adalah data jamak memiliki label *coreference* lebih dari satu angka karena kata ganti kepemilikan orang jamak seperti “kita”, “kami”, “kalian”, “mereka” dapat merujuk ke beberapa entitas berbeda. Untuk lebih jelasnya, berikut merupakan contoh data hasil pelabelan dari teks novel Sherlock Holmes.

Tabel 3.9 Data hasil pelabelan

Lokasi dokumen	Lokasi kalimat	Lokasi token	Token	POS tag	NER	Label
1	1	1	Bagi	IN	O	
1	1	2	Sherlock Holmes	NNP	B- PERSON	1;1;2;1
1	1	3	,	Z	O	
1	1	4	dia	PRP	O	1;1;4;2
1	1	5	adalah	VB	O	
1	1	6	wanita	NN	O	
1	1	7	yang	SC	O	
1	1	8	istimewa	JJ	O	
1	1	9	.	Z	O	
1	2	10	Dia	PRP	O	1;2;10;1
1	2	11	tak	NEG	O	
1	2	12	pernah	NN	O	
1	2	13	menyebut	VB	O	
1	2	14	wanita	NN	O	
1	2	15	itu	PR	O	
1	2	16	dengan	IN	O	
1	2	17	istilah	NN	O	
1	2	18	lain	JJ	O	
1	2	19	.	Z	O	
1	3	20	Di	IN	O	

Tabel 3.9 Data hasil pelabelan (lanjutan)

Lokasi dokumen	Lokasi kalimat	Lokasi kata/token	Token	POS tag	NER	Label
1	3	21	mata	NN	O	
1	3	22	nya	PRP	O	1;3;22;1
1	3	23	wanita	NN	O	
1	3	24	itulah	PR	O	
1	3	25	yang	SC	O	
1	3	26	paling	RB	O	
1	3	27	hebat	JJ	O	
1	3	28	di	IN	O	
1	3	29	antara	NN	O	
1	3	30	seluruh	CD	O	
1	3	31	kaum	NN	O	
1	3	32	nya	PRP	O	1;3;32;2
1	3	33	.	Z	O	
1	4	34	Ini	PR	O	
1	4	35	tidak	NEG	O	
1	4	36	berarti	VB	O	
1	4	37	bahwa	SC	O	
1	4	38	Holmes	NNP	U- PERSON	1;4;38;1
1	4	39	mencintai	VB	O	
1	4	40	Irene Adler	NNP	B- PERSON	1;4;40;2
1	4	41	.	Z	O	

Banyaknya token yang diberi label *coreference* pada data *train* adalah sebanyak 1501 token dari 13939 total seluruh token. Pelabelan pada data penelitian ini dilakukan secara manual oleh peneliti serta dikoreksi dan divalidasi oleh seorang ahli linguistik yang paham akan permasalahan dalam penelitian ini. Adapun biodata beliau adalah sebagai berikut.

1. Nama Lengkap : Agesti Siwi Prabantari
2. Umur : 26 Tahun
3. Aktivitas : Mahasiswa Magister Sastra
4. Instansi : Universitas Gadjah Mada
5. Alamat : Jalan Parangtritis, Panggungharjo, Sewon, Bantul
6. Pengalaman
 - : 1. Copywriter (2021)
 - Instagram Yayasan Sanggar Inovasi Desa
 - Tiktok Yayasan Sanggar Inovasi Desa
 - Twitter Yayasan Sanggar Inovasi Desa
 2. Content Writer (2021)
 - Yayasan Sanggar Inovasi Desa (ysid.or.id)
 3. Editor (2021—2022)
 - CV. Karsa Pandiva Media

3.5. Ekstraksi Fitur

Pada proses ekstraksi fitur, teks atau data teks diubah menjadi representasi numerik yang dapat dipahami oleh metode pembelajaran mesin sebagai inputan. Proses ini dilakukan setelah melalui tahapan *pre-processing* sebelumnya. Pada penelitian ini, semua kemungkinan pasangan *coreference* akan dicoba dipasangkan sesuai dengan tiga kondisi pada pelabelan *coreference* sebelumnya, dimana mengikuti aturan bahwa posisi baris token i lebih besar dari posisi baris token j ($i > j$).

Ekstraksi fitur akan memasangkan setiap token i dan token j yang memiliki pelabelan *coreference* yang berbeda. Mengacu pada data di Tabel 3.9, contohnya adalah token **Sherlock Holmes** memiliki label **1;1;2;1** dan **dia** memiliki label **1;1;4;2**, karena dua token tersebut memiliki label yang berbeda maka **Sherlock Holmes** dan **dia** akan dijadikan pasangan i dan j . Dimana i adalah **dia** dan j adalah **Sherlock Holmes** karena posisi baris token i harus berada setelah token j ($i > j$). Adapun fitur-fitur yang digunakan pada penelitian ini adalah sebagai berikut.

Tabel 3.10 Ekstraksi fitur

Fitur untuk mendeskripsikan token -i	
<i>novel_id_i</i>	Mendeskripsikan nilai <i>id</i> dokumen dimana token <i>i</i> berada
<i>sentence_id_i</i>	Mendeskripsikan nilai <i>id</i> kalimat dimana token <i>i</i> berada
<i>word_id_i</i>	Mendeskripsikan nilai <i>id</i> dimana token <i>i</i> berada
<i>is_pronoun_i</i>	Mendeskripsikan apakah nilai POS tag token <i>i</i> adalah PRP. Jika benar maka bernilai 1, jika tidak bernilai 0.
<i>is_propernoun_i</i>	Mendeskripsikan apakah nilai POS tag token <i>i</i> adalah NNP. Jika benar maka bernilai 1, jika tidak bernilai 0.
<i>is_plural_i</i>	Mendeskripsikan apakah token <i>i</i> termasuk kata ganti kepemilikan jamak (“kita”, “kami”, “kalian”, “mereka”). Jika benar maka bernilai 1, jika tidak bernilai 0.
<i>is_singular_i</i>	Mendeskripsikan apakah token <i>i</i> termasuk kata ganti kepemilikan tunggal (“saya”, “aku”, “ku”, “engkau”, “kamu”, “anda”, “kau”, “mu”, “dia”, “ia”, “beliau”, “nya”). Jika benar maka bernilai 1, jika tidak bernilai 0.
<i>is_ner_not_o_i</i>	Mendeskripsikan apakah nilai NER token <i>i</i> bukan O. Jika benar maka bernilai 1, jika tidak bernilai 0.
Fitur untuk mendeskripsikan token -j	
<i>novel_id_j</i>	Mendeskripsikan nilai <i>id</i> dokumen dimana token <i>j</i> berada
<i>sentence_id_j</i>	Mendeskripsikan nilai <i>id</i> kalimat dimana token <i>j</i> berada
<i>word_id_j</i>	Mendeskripsikan nilai <i>id</i> dimana token <i>j</i> berada
<i>is_pronoun_j</i>	Mendeskripsikan apakah nilai POS tag token <i>j</i> adalah PRP. Jika benar maka bernilai 1, jika tidak bernilai 0.
<i>is_propernoun_j</i>	Mendeskripsikan apakah nilai POS tag token <i>j</i> adalah NNP. Jika benar maka bernilai 1, jika tidak bernilai 0.
<i>is_plural_j</i>	Mendeskripsikan apakah token <i>j</i> termasuk kata ganti kepemilikan jamak (“kita”, “kami”, “kalian”, “mereka”). Jika benar maka bernilai 1, jika tidak bernilai 0.

Tabel 3.10 Ekstraksi fitur (lanjutan)

Fitur untuk mendeskripsikan token -j	
<i>is_singular_j</i>	Mendeskripsikan apakah token <i>j</i> termasuk kata ganti kepemilikan tunggal (“saya”, “aku”, “ku”, “engkau”, “kamu”, “anda”, “kau”, “mu”, “dia”, “ia”, “beliau”, “nya”). Jika benar maka bernilai 1, jika tidak bernilai 0.
<i>is_ner_not_o_j</i>	Mendeskripsikan apakah nilai NER token <i>j</i> bukan O. Jika benar maka bernilai 1, jika tidak bernilai 0.
Fitur untuk relasi antara token-i dan token-j	
<i>distance_of_word</i> (DOW)	Mendeskripsikan jarak antar token <i>i</i> dan <i>j</i> . Berikut rumus untuk mendapatkan nilai DOW:
	$DOW = \frac{\text{jarak token } i - \text{jarak token } j}{\text{jumlah token dalam dataset}}$
<i>distance_of_sentence</i> (DOS)	Mendeskripsikan jarak antar kalimat token <i>i</i> dan token <i>j</i> . Berikut rumus untuk mendapatkan nilai DOS:
	$DOS = \frac{\text{jarak kalimat token } i - \text{jarak kalimat token } j}{\text{jumlah kalimat dalam dataset}}$
<i>is_string_match</i>	Mendeskripsikan <i>string</i> token <i>i</i> dan <i>j</i> , jika <i>string</i> token <i>i</i> dan <i>j</i> sama maka bernilai 1, jika berbeda maka bernilai 0.
<i>alias</i>	Mendeskripsikan potongan <i>string</i> (<i>substring</i>) token <i>i</i> dan <i>j</i> , jika <i>substring</i> dari token <i>i</i> berada dalam token <i>j</i> dan sebaliknya maka bernilai 1, jika tidak maka bernilai 0.

Berikut adalah contoh hasil dari penerapan ekstraksi fitur pada data *train* pada Tabel 3.11.

Tabel 3.11 Penerapan proses ekstraksi fitur

Token <i>i</i>	Token <i>j</i>	Fitur <i>i</i>							Fitur <i>j</i>							Fitur Relasi					
		novel_id_j	sentence_id_j	word_id_j	is_pronoun_i	is_propernoun_i	is_plural_i	is_singular_i	is_ner_not_o_i	novel_id_j	sentence_id_j	word_id_j	is_pronoun_j	is_propernoun_:	is_plural_j	is_singular_j	is_ner_not_o_j	dow	dos	is_string_match	alias
dia	Sherlock Holmes	1	1	4	1	0	0	1	0	1	1	2	0	1	0	0	1	0,0001	0	0	0
Dia	Sherlock Holmes	1	2	10	1	0	0	1	0	1	1	2	0	1	0	0	1	0,0006	0,0008	0	0
Dia	dia	1	2	10	1	0	0	1	0	1	1	4	1	0	0	1	0	0,0004	0,0008	1	0
nya	Sherlock Holmes	1	3	22	1	0	0	1	0	1	1	2	0	1	0	0	1	0,0014	0,0015	0	0
nya	dia	1	3	22	1	0	0	1	0	1	1	4	1	0	0	1	0	0,0013	0,0015	0	0
nya	Dia	1	3	22	1	0	0	1	0	1	2	10	1	0	0	1	0	0,0009	0,0008	0	0
nya	Sherlock Holmes	1	3	32	1	0	0	1	0	1	1	2	0	1	0	0	1	0,0021	0,0015	0	0
nya	dia	1	3	32	1	0	0	1	0	1	1	4	1	0	0	1	0	0,002	0,0015	0	0

Tabel 3.11 Penerapan proses ekstraksi fitur (lanjutan)

Token <i>i</i>	Token <i>j</i>	Fitur <i>i</i>							Fitur <i>j</i>							Fitur Relasi					
		novel_id_j	sentence_id_j	word_id_i	is_pronoun_i	is_propernoun_i	is_plural_i	is_singular_i	is_ner_not_o_i	novel_id_j	sentence_id_j	word_id_j	is_pronoun_j	is_propernoun_j	is_plural_j	is_singular_j	is_ner_not_o_j	dow	dos	is_string_match	alias
nya	Dia	1	3	32	1	0	0	1	0	1	2	10	1	0	0	1	0	0,0016	0,0008	0	0
nya	nya	1	3	32	1	0	0	1	0	1	3	22	1	0	0	1	0	0,0007	0	1	0
Holmes	Sherlock Holmes	1	4	38	0	1	0	0	1	1	1	2	0	1	0	0	1	0,0026	0,0023	0	1
Holmes	dia	1	4	38	0	1	0	0	1	1	1	4	1	0	0	1	0	0,0024	0,0023	0	0
Holmes	Dia	1	4	38	0	1	0	0	1	1	2	10	1	0	0	1	0	0,002	0,0015	0	0
Holmes	nya	1	4	38	0	1	0	0	1	1	3	22	1	0	0	1	0	0,0011	0,0008	0	0
Holmes	nya	1	4	38	0	1	0	0	1	1	3	32	1	0	0	1	0	0,0004	0,0008	0	0
Irene Adler	Sherlock Holmes	1	4	40	0	1	0	0	1	1	1	2	0	1	0	0	1	0,0027	0,0023	0	0
Irene Adler	dia	1	4	40	0	1	0	0	1	1	1	4	1	0	0	1	0	0,0026	0,0023	0	0

3.6. Proses Training

Setelah data *train* maupun data validasi melalui proses ekstraksi fitur dan pelabelan, tahapan berikutnya adalah tahapan *training* model menggunakan metode *Random Forest Classifier*. Namun, sebelum memulai proses *training* model, data hasil proses ekstraksi fitur dan pelabelan harus digabungkan terlebih dahulu agar dapat digunakan sebagai data masukan untuk proses *training*. Berikut adalah format komponen *dataset* yang digunakan untuk data masukan *training* model menggunakan metode *Random Forest Classifier*:

1. Fitur (*features*)

Fitur-fitur *dataset* merupakan variabel-variabel independen yang digunakan untuk memprediksi label atau target. Setiap baris dalam *dataset* akan memiliki nilai-nilai fitur yang terkait dengan sampel. Fitur-fitur ini dapat berupa data numerik dan kategorikal. Dalam penelitian ini, fitur-fitur yang digunakan merupakan data hasil ekstraksi fitur (lihat pada Tabel 3.10) yang terdiri dari 20 fitur yaitu *novel_id_i*, *sentence_id_i*, *word_id_i*, *is_pronoun_i*, *is_propernoun_i*, *is_plural_i*, *is_singular_i*, *is_ner_not_o_i*, *novel_id_j*, *sentence_id_j*, *word_id_j*, *is_pronoun_j*, *is_propernoun_j*, *is_plural_j*, *is_singular_j*, *is_ner_not_o_j*, *dow*, *dos*, *is_string_match*, *alias*.

2. Label (target)

Label atau target adalah variabel dependen yang ingin diprediksi oleh model. Label ini bisa berupa kelas biner (dua kelas: misalnya, “0” dan “1”) atau kelas multi-label. Pada penelitian ini label yang digunakan adalah label positif dan negatif (1 dan -1). Label positif digunakan untuk pasangan yang merupakan *coreference* sedangkan label negatif digunakan untuk pasangan bukan *coreference*. Data label dalam penelitian ini didapatkan dari hasil pelabelan yang dilakukan sebelumnya. Dapat dilihat seperti pada Tabel 3.9 kolom “Label” terdapat angka-angka yang mewakili label pada tiap tokennya. Label *coreference* yang terletak pada angka keempat dan seterusnya menjadi penanda apakah tiap token *i* dan token *j* yang dipasangkan memiliki angka keempat dan seterusnya sama antar satu sama lain. Jika benar maka pasangan token tersebut adalah *coreference* dan bernilai label positif (1) dan jika tidak maka bukan *coreference* dan bernilai label negatif (-1).

Berikut merupakan *dataset* hasil penggabungan fitur dan label yang akan digunakan untuk data masukan dalam proses *training*.

Tabel 3.12 Data fitur dan label

Fitur (20)	Label
1 1 4 1 0 0 1 0 1 1 2 0 1 0 0 1 0,00014 0,0 0 0	1
1 2 10 1 0 0 1 0 1 1 2 0 1 0 0 1 0,00056 0,00076	1
0 0	
1 2 10 1 0 0 1 0 1 1 4 1 0 0 1 0 0,00042 0,00077	-1
1 0	
1 3 22 1 0 0 1 0 1 1 2 0 1 0 0 1 0,00142 0,00154	1
0 0	
1 3 22 1 0 0 1 0 1 1 4 1 0 0 1 0 0,00128 0,00154	-1
0 0	
1 3 22 1 0 0 1 0 1 2 10 1 0 0 1 0 0,00085 0,00077	1
0 0	
1 3 32 1 0 0 1 0 1 1 2 0 1 0 0 1 0,00213 0,00154	1
0 0	
.	.
.	.
.	.
18 1293 13924 1 0 1 0 0 18 1292 13916 1 0 1 0 0	1
0,00056 0,00077 0 0	
18 1293 13924 1 0 1 0 0 18 1292 13920 0 1 0 0 1	1
0,00028 0,00077 0 0	

Jumlah keseluruhan data fitur dan label pada Tabel 3.12 adalah sebanyak 109306 baris data. Dimana data tersebut selanjutkan akan dibagi menjadi dua bagian yaitu data *train* dan data validasi. Data *train* untuk melatih model dan data validasi untuk validasi model yang telah dilatih untuk mencegah terjadinya *overfitting*. Porsi pembagiannya adalah 87444 pasangan *coreference* (80%) untuk data *train* dan 21862 pasangan (20%) untuk data validasi.

Langkah selanjutnya adalah proses *training* dengan menggunakan metode *Random Forest Classifier*. Metode ini bekerja dengan menggabungkan prediksi dari beberapa model (disebut sebagai pohon keputusan) untuk meningkatkan kinerja prediksi keseluruhan. Setiap pohon dipelajari pada *subset* acak dari data, dan kemudian hasil prediksi dari semua pohon digabungkan menjadi prediksi akhir.

Dalam penelitian ini juga digunakan *RandomSearchCV* dimana metode ini digunakan untuk mencari kombinasi *hyperparameter* terbaik untuk model. Pencarian dilakukan dengan mengeksplorasi ruang parameter dengan memilih kombinasi parameter secara acak dari distribusi yang telah ditentukan. Banyaknya kombinasi ditentukan dengan nilai iterasi yang diinginkan. Untuk setiap iterasi, satu set kombinasi *hyperparameter* akan dipilih secara acak.

Cara kerja gabungan metode *Random Forest Classifier* dan *RandomSearchCV* adalah sebagai berikut.

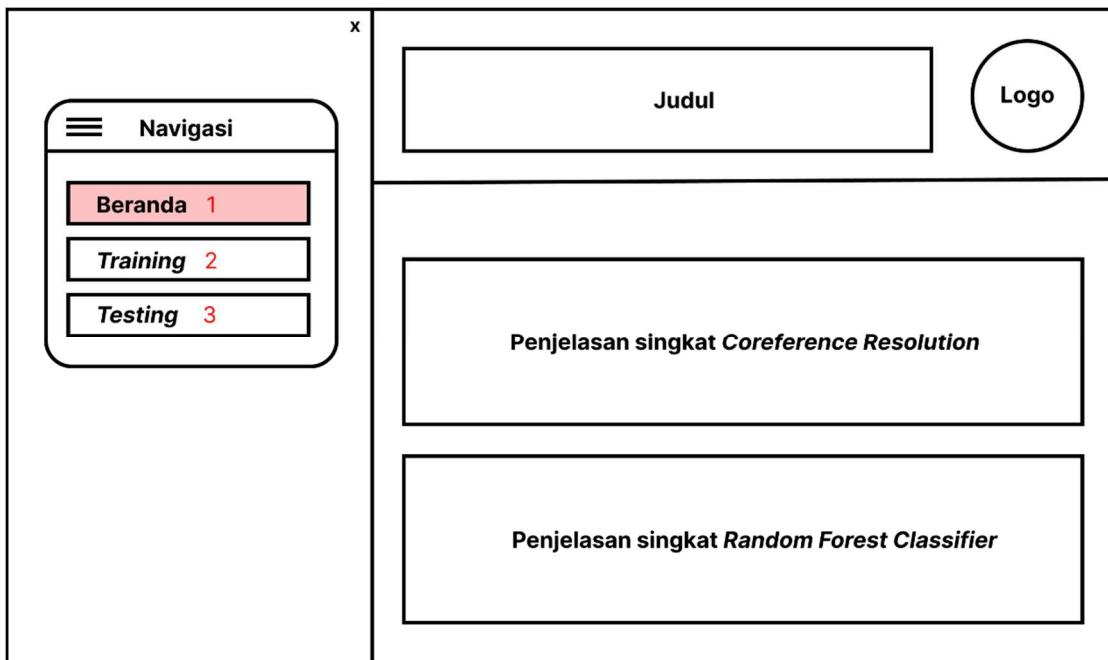
1. Inisialisasi model, yaitu dimana objek *Random Forest Classifier* dibuat dengan memberikan setiap parameter default atau nilai awal.
2. Menentukan rentang *hyperparameter* yang akan dijelajahi oleh *RandomSearchCV*. *Hyperparameter* metode *Random Forest Classifier* yang digunakan adalah *n_estimators*, *max_depth*, *max_features*, *min_samples_leaf*, *min_samples_split*, dan *bootstrap*.
3. Pembuatan objek *RandomSearchCV* dengan menyertakan model *Random Forest Classifier* yang telah diinisialisasi, rentang *hyperparameter*, skema penilaian (misalnya validasi silang), dan jumlah iterasi (jumlah kombinasi *hyperparameter* yang akan diuji).
4. Pencarian *hyperparameter* dimana *RandomSearchCV* akan mencoba kombinasi *hyperparameter* secara acak dari rentang yang telah ditentukan. Untuk setiap kombinasi *hyperparameter* yang dipilih secara acak, *RandomSearchCV* akan melatih dan mengevaluasi model menggunakan *cross-validation*. Model akan dilatih dan diuji pada beberapa *subset* data yang berbeda.
5. Setelah selesai mencoba semua kombinasi, *RandomSearchCV* akan memberikan hasil yang berisi kombinasi *hyperparameter* terbaik beserta skor evaluasi yang terkait.

3.8. Perancangan Sistem

Untuk mempermudah pengguna dalam menjalankan sistem, peneliti menggunakan sistem berbasis web sebagai tampilan sistem dengan pengguna. Rancangan sistem untuk penelitian ini adalah halaman beranda, halaman *training*, dan halaman *testing*.

3.8.1. Rancangan tampilan halaman beranda

Halaman ini adalah halaman pertama yang akan muncul ketika sistem dijalankan. Pada halaman ini berisi informasi mengenai judul penelitian, penjelasan singkat mengenai *Coreference Resolution* serta metode *Random Forest Classifier*, dan tombol untuk terhubung ke halaman lain pada *sidebar*. Rancangan tampilan halaman Beranda dapat dilihat pada Gambar 3.12.



Gambar 3.12 Rancangan halaman Beranda

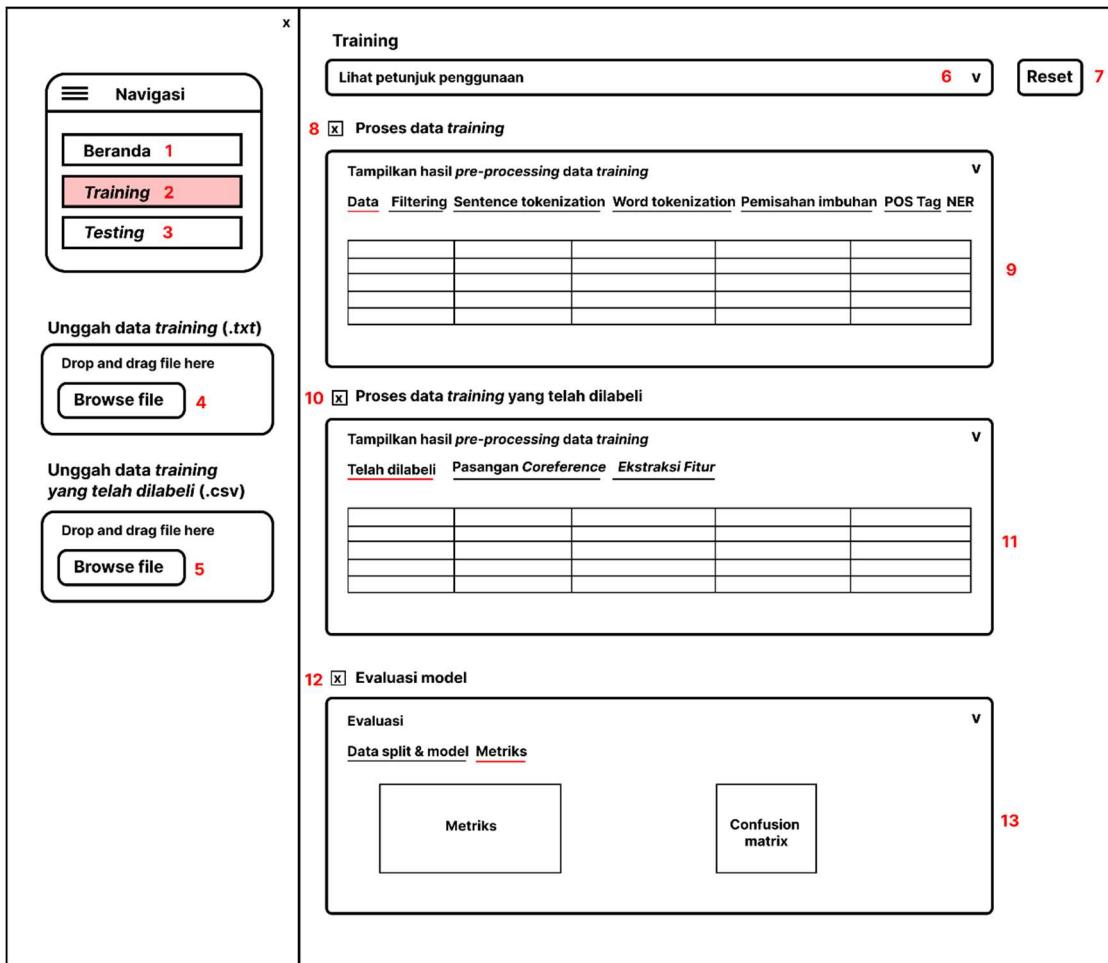
Keterangan:

1. Elemen bermotor 1 adalah tombol untuk mengarahkan ke halaman Beranda.
2. Elemen bermotor 2 adalah tombol untuk mengarahkan ke halaman *Training*.
3. Elemen bermotor 3 adalah tombol untuk mengarahkan ke halaman *Testing*.

3.8.2. Rancangan tampilan halaman training

Pada tampilan halaman *Training* akan ditampilkan elemen *expander* yang ketika diklik akan menampilkan cara penggunaan halaman *Training* secara detail. Pada *sidebar* sebelah kiri terdapat dua tombol untuk mengunggah file data *train* mentah dan data

yang sudah dilabeli. Terdapat pula tiga elemen *checkbox* untuk memproses data *train*, memproses data *train* yang sudah dilabeli, dan evaluasi model. Di sebelah kanan halaman terdapat tombol *reset* untuk mengembalikan halaman ke mode sebelum *checkbox* dicentang. Rancangan tampilan halaman *Training* dapat dilihat pada Gambar 3.13.



Gambar 3.13 Rancangan halaman *Training*

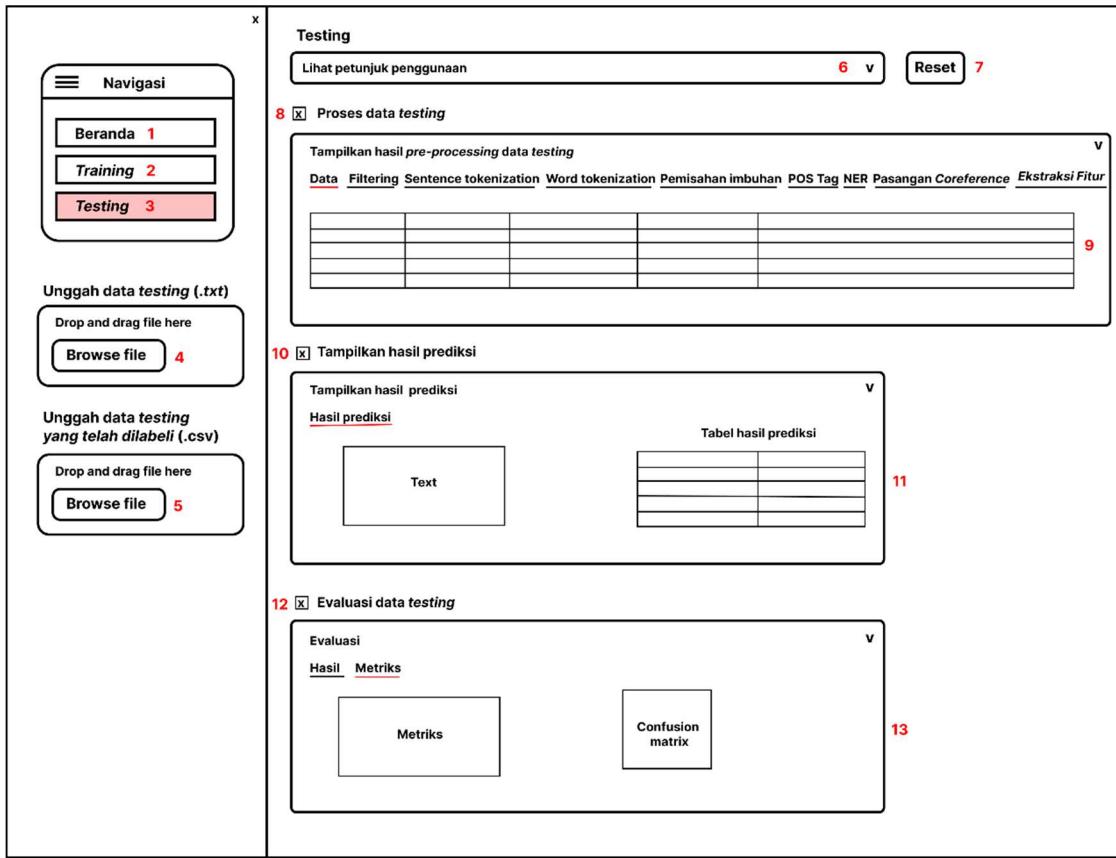
Keterangan:

1. Elemen bernomor 1 adalah tombol untuk mengarahkan ke halaman Beranda.
2. Elemen bernomor 2 adalah tombol untuk mengarahkan ke halaman *Training*.
3. Elemen bernomor 3 adalah tombol untuk mengarahkan ke halaman *Testing*.
4. Elemen bernomor 4 adalah tombol untuk menunggah file data *train* mentah dengan ekstensi *.txt*.
5. Elemen bernomor 5 adalah tombol untuk mengunggah file data *train* yang sudah dilabeli dengan ekstensi *.csv*.

6. Elemen bernomor 6 adalah *expander* yang didalamnya merupakan kontainer berisi petunjuk penggunaan aplikasi.
7. Elemen bernomor 7 adalah tombol untuk mengembalikan halaman *training* seperti semula.
8. Elemen bernomor 8 adalah *checkbox* yang ketika diklik akan memproses data *train* dan menampilkan *expander* berisi hasil proses.
9. Elemen bernomor 9 adalah kontainer yang menampilkan hasil *pre-processing* dimulai dari data teks sampai dengan proses NER.
10. Elemen bernomor 10 adalah *checkbox* yang ketika diklik akan memproses data yang sudah dilabeli dan menampilkan *expander* yang berisi hasil proses.
11. Elemen bernomor 11 adalah kontainer yang menampilkan hasil proses mulai dari data yang dilabeli sampai dengan proses ekstraksi fitur.
12. Elemen bernomor 12 adalah *checkbox* yang ketika diklik akan menampilkan hasil evaluasi data validasi.
13. Elemen bernomor 13 adalah kontainer yang menampilkan rincian model serta hasil evaluasi data validasi berupa metriks dan *confusion matrix*.

3.8.3. Rancangan tampilan halaman testing

Pada tampilan halaman *Testing* akan ditampilkan elemen *expander* yang ketika diklik akan menampilkan cara penggunaan halaman *Testing* secara detail. Pada sidebar sebelah kiri terdapat dua tombol untuk mengunggah file data *test* mentah dan data yang sudah dilabeli. Terdapat pula tiga elemen *checkbox* untuk memproses data *test*, menampilkan hasil prediksi, dan hasil evaluasi. Di sebelah kanan halaman terdapat tombol reset untuk mengembalikan halaman ke mode sebelum checkbox dicentang. Rancangan tampilan halaman *Testing* dapat dilihat pada Gambar 3.14.



Gambar 3.14 Rancangan halaman *Testing*

Keterangan:

1. Elemen bernomor 1 adalah tombol untuk mengarahkan ke halaman Beranda.
2. Elemen bernomor 2 adalah tombol untuk mengarahkan ke halaman *Training*.
3. Elemen bernomor 3 adalah tombol untuk mengarahkan ke halaman *Testing*.
4. Elemen bernomor 4 adalah tombol untuk menungguh file data *test* mentah dengan ekstensi *.txt*.
5. Elemen bernomor 5 adalah tombol untuk mengungguh file data *test* yang sudah dilabeli dengan ekstensi *.csv*.
6. Elemen bernomor 6 adalah *expander* yang didalamnya merupakan kontainer berisi petunjuk penggunaan aplikasi.
7. Elemen bernomor 7 adalah tombol untuk mengembalikan halaman *testing* seperti semula.
8. Elemen bernomor 8 adalah *checkbox* yang ketika diklik akan memproses data *test* dan menampilkan *expander* berisi hasil proses.

9. Elemen bernomor 9 adalah kontainer yang menampilkan hasil *pre-processing* dimulai dari data teks sampai dengan proses ekstraksi fitur.
10. Elemen bernomor 10 adalah *checkbox* yang ketika diklik akan menampilkan *expander* yang berisi hasil prediksi data *test*.
11. Elemen bernomor 11 adalah kontainer yang menampilkan hasil prediksi berupa teks dan tabel prediksi.
12. Elemen bernomor 12 adalah *checkbox* yang ketika diklik akan menampilkan hasil evaluasi data *test*.
13. Elemen bernomor 13 adalah kontainer yang menampilkan hasil evaluasi data *test* berupa metriks dan *confusion matrix*.

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi Sistem

Detail penggunaan perangkat keras dan perangkat lunak dalam pembuatan sistem *coreference resolution* ini adalah sebagai berikut.

4.1.1. Spesifikasi perangkat keras dan perangkat lunak

Spesifikasi perangkat keras yang digunakan untuk membangun sistem adalah sebagai berikut.

Tabel 4.1 Spesifikasi perangkat keras

No	Perangkat Keras	Spesifikasi
1	<i>Processor</i>	<i>Intel® Core™ i5-7200U CPU @ 2.50GHz 2.71 GHz</i>
2	Memori (RAM)	12 GB
3	<i>Hard disk</i>	1 TB

Spesifikasi perangkat lunak yang digunakan untuk membangun sistem adalah sebagai berikut.

Tabel 4.2 Spesifikasi perangkat lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem operasi	<i>Microsoft Windows 10 Enterprise 64-bit</i>
2	<i>Code editor</i>	<i>Microsoft Visual Studio Code</i>
3	Bahasa pemrograman & <i>framework</i>	- <i>Python</i> versi 3.8.10 - <i>Streamlit</i> versi 1.22.0

Tabel 4.2 Spesifikasi perangkat lunak (lanjutan)

No	Perangkat Lunak	Spesifikasi
4	<i>Python library</i>	<ul style="list-style-type: none"> - <i>scikit-learn</i> versi 1.2. - <i>NLTK</i> versi 3.7 - <i>numpy</i> versi 1.23.1 - <i>pandas</i> versi 1.4.3 - <i>scipy</i> versi 1.10.1 - <i>seaborn</i> versi 0.12.1 - <i>matplotlib</i> - <i>sklearn_crfsuite</i>.
5	Diagram	<i>Draw.io</i>

4.1.2. Implementasi perancangan tampilan antarmuka

Implementasi dari rancangan tampilan antarmuka yang telah dijelaskan pada Bab 3 akan diuraikan sebagai berikut.

1. Tampilan halaman Beranda

Halaman Beranda adalah halaman awal untuk pengoperasian sistem. Halaman ini berisi informasi mengenai judul penelitian, penjelasan singkat mengenai *Coreference Resolution* serta metode *Random Forest Classifier*, dan *button* untuk terhubung ke halaman lain yaitu halaman *Training* dan *Testing* pada *sidebar*. Tampilan halaman Beranda dapat dilihat pada Gambar 4.1.

Coreference Resolution Untuk Teks Bahasa Indonesia Menggunakan Random Forest Classifier

Apa itu Coreference Resolution?

Coreference resolution merupakan salah satu tugas penting dalam bidang pengolahan bahasa alami atau NLP (Natural Language Processing). Tugas ini berkaitan dengan identifikasi dan penyelesaian referensi yang merujuk pada entitas yang sama dalam teks.

Coreference resolution biasanya digunakan untuk mendukung pengembangan tugas-tugas lain dalam bidang NLP seperti sistem dialog, pemahaman teks, terjemahan mesin, dan pemrosesan teks otomatis.

Apa itu Random Forest Classifier?

Random Forest Classifier adalah salah satu jenis algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi. Ini adalah algoritma ensemble yang menggabungkan beberapa pohon keputusan (decision trees) untuk membuat prediksi yang lebih akurat.

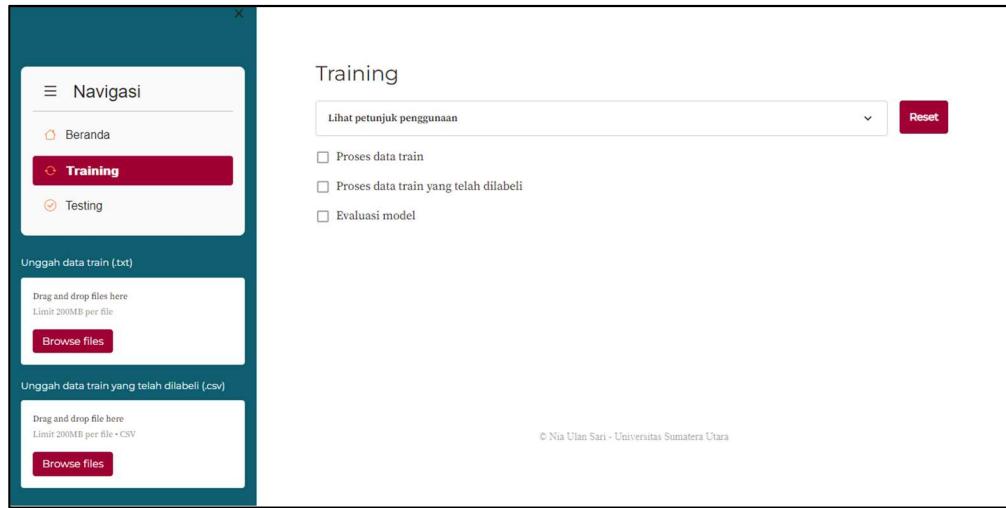
Random Forest mengambil nama dari konsep hutan (forest) karena algoritma ini menggabungkan banyak pohon keputusan (trees) ke dalam satu model yang kuat.

© Nia Ulan Sari - Universitas Sumatera Utara

Gambar 4.1 Tampilan halaman Beranda

2. Tampilan halaman *Training*

Pada halaman *Training* terdapat beberapa elemen diantaranya adalah *expander* yang berisi kontainer penjelasan petunjuk penggunaan halaman *Training*. Kemudian terdapat tombol “*Reset*” untuk mengembalikan keadaan halaman ke seperti semula. Kemudian dua tombol untuk menunggu data *train* mentah dan data *train* yang sudah dilabeli. Tiga *checkbox* yang ketika diklik akan menjalankan proses *training* dimulai dari *pre-processing* sampai hasil evaluasi model. Tampilan awal halaman *Training* dapat dilihat pada Gambar 4.2.



Gambar 4.2 Tampilan awal halaman *Training*

Memulai proses *training* diawali dengan mengunggah data *train* dengan ekstensi *.txt* dengan mengklik tombol “*Browse files*” pertama pada sidebar kiri. Data *train* yang digunakan pada penelitian ini adalah sebanyak 18 *file* teks novel. Setelah *file-file* tersebut diunggah, selanjutnya adalah centang *checkbox* “*Proses data train*” untuk menampilkan hasil *pre-processing* data mulai dari data teks sampai dengan proses NER. Pengguna dapat mengklik salah satu *tab* proses diantara *tab* data, *filtering*, *sentence tokenization*, *word tokenization*, pemisahan imbuhan, POS tag, dan NER untuk melihat hasil masing-masing tahapan. Berikut tampilan halaman *Training* yang menampilkan data teks dan hasil akhir proses NER dapat dilihat pada Gambar 4.3 dan Gambar 4.4.

ID	Novel ID	Text
1		Bagi Sherlock Holmes, dia adalah wanita yang istimewa. Dia tak pernah menyebut wanita itu dengan istilah lain. Dia
2		Ray Hirano bersiul pelan sambil melihat ke kiri dan ke kanan sebelum berjalan cepat menyeberangi jalan ke arah sa
3		Jakarta, Desember 1991. Pada suatu malam yang terasa lebih ringan dari malam-malam sebelumnya bagi seorang pr
4		Alva dan Danny tinggal di satu kamar. Sementara Lydia dan Etta tinggal di kamar masing-masing yang manggil. Aku r
5		“Sekarang aku masih di jalan...baru pulang kantor... Aku juga tahu sekarang sudah jam sepuluh... Ya, jam sepuluh le
6		Jimbron yang tambun dan invalid kakinya panjang sebelah terengah-engah di belakangku. Wajahnya pias. Dahinya y
7		Ruangan 4 x 4 m ² itu selintas terlihat terlalu sederhana untuk sebuah ruangan paling mutakhir di kota ini. I
8		Adzan shubuh dari meunasher terdengar syahdu. Bersahutan satu sama lain. Menggentarkan langit-langit Lhok Nga y
9		Matilah engkau mati. Kau akan lahir berkali-kali....
10		Gerimis membungkus halaman sekolah. Langit mendung. Gumpalan awan hitam seakan bosan beranjak di atas san

Gambar 4.3 Tampilan halaman *Training* yang menampilkan tabel data teks

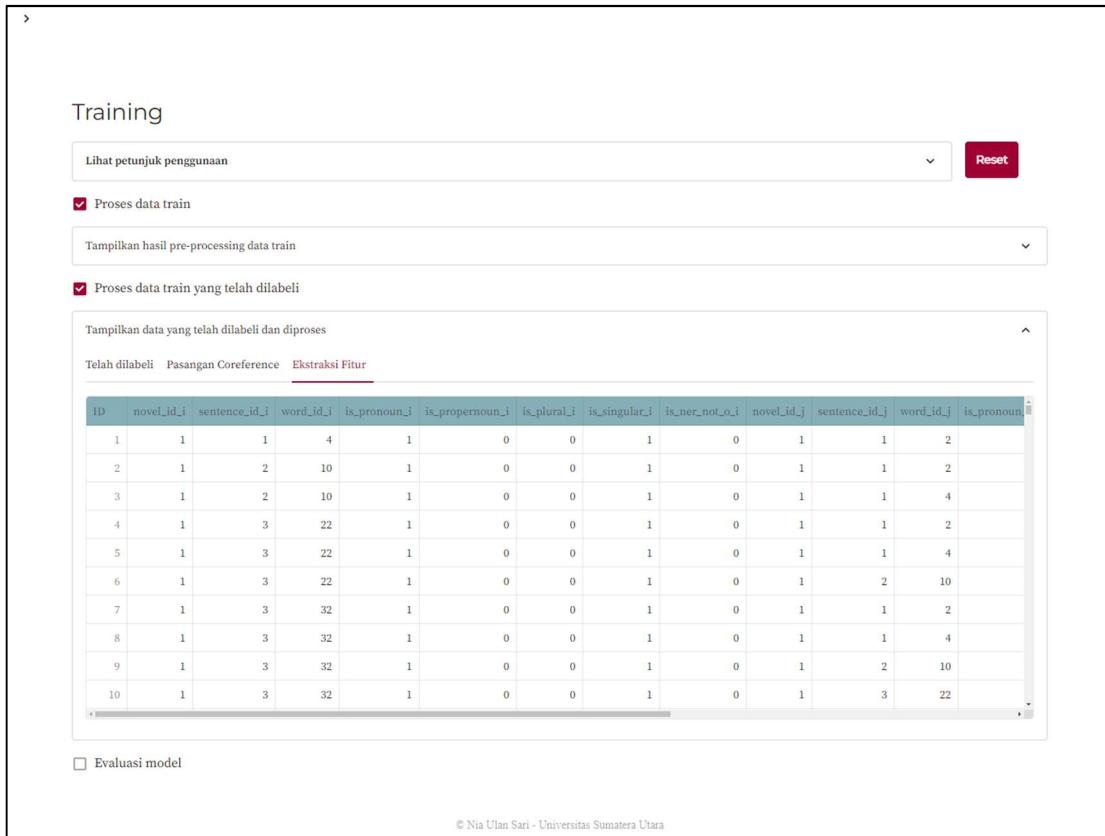
The screenshot shows a user interface for training a Natural Language Processing model. At the top, there's a button to 'Lihat petunjuk penggunaan' (View user guide) and a 'Reset' button. Below that, a checkbox 'Proses data train' (Process training data) is checked. A link 'Unduh data yang akan dilabeli' (Download data to be labeled) is circled in red. The main area displays a table titled 'Tampilkan hasil pre-processing data train' (Show pre-processing results of training data). The table has columns: ID, Novel ID, Sentence ID, Word ID, Word, and NER. The NER column uses standard BIO tags. The table data is as follows:

ID	Novel ID	Sentence ID	Word ID	Word	NER
1		1	1	Bagi	O
2		1	1	Sherlock Holmes	B-PERSON
3		1	1	,	O
4		1	1	dia	O
5		1	1	adalah	O
6		1	1	wanita	O
7		1	1	yang	O
8		1	1	istimewa	O
9		1	1	.	O
10		1	2	Dia	O

At the bottom, there are two checkboxes: 'Proses data train yang telah dilabeli' (Process training data that has been labeled) and 'Evaluasi model' (Evaluate model).

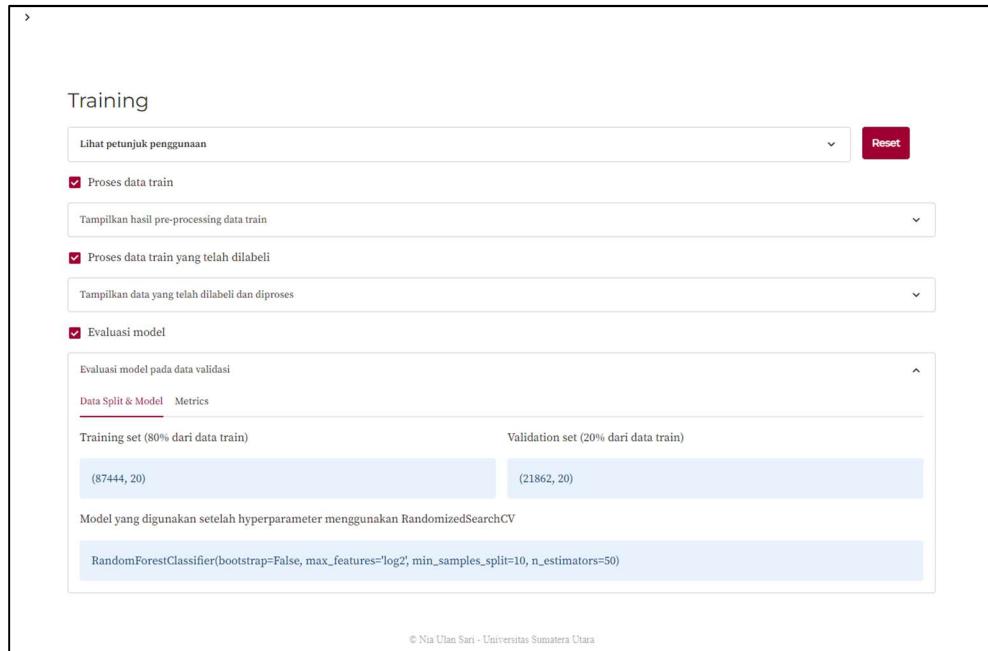
Gambar 4.4 Tampilan halaman Training yang menampilkan tabel hasil proses NER

Selanjutnya pengguna dapat mengunduh data hasil akhir *pre-processing* untuk diberi label dengan mengklik link “Unduh data yang akan dilabeli” seperti yang tertera pada Gambar 4.4 (lingkar merah). Setelah data diunduh, pengguna dapat memberikan label *coreference* pada data tersebut dan lanjut ke proses selanjutnya dengan mengunggah data yang sudah dilabeli tersebut dengan mengklik tombol “*Browse files*” kedua pada *sidebar* kiri dan mencentang *checkbox* “Proses data *train* yang telah dilabeli” untuk menampilkan data yang telah berlabel, pasangan *coreference*, dan hasil ekstraksi fitur. Pengguna dapat mengklik salah satu *tab* proses diantara *tab* data berlabel, pasangan *coreference*, dan ekstraksi fitur untuk melihat hasil masing-masing tahapan. Tampilan halaman *Training* yang menampilkan hasil ekstraksi fitur dapat dilihat pada Gambar 4.5.

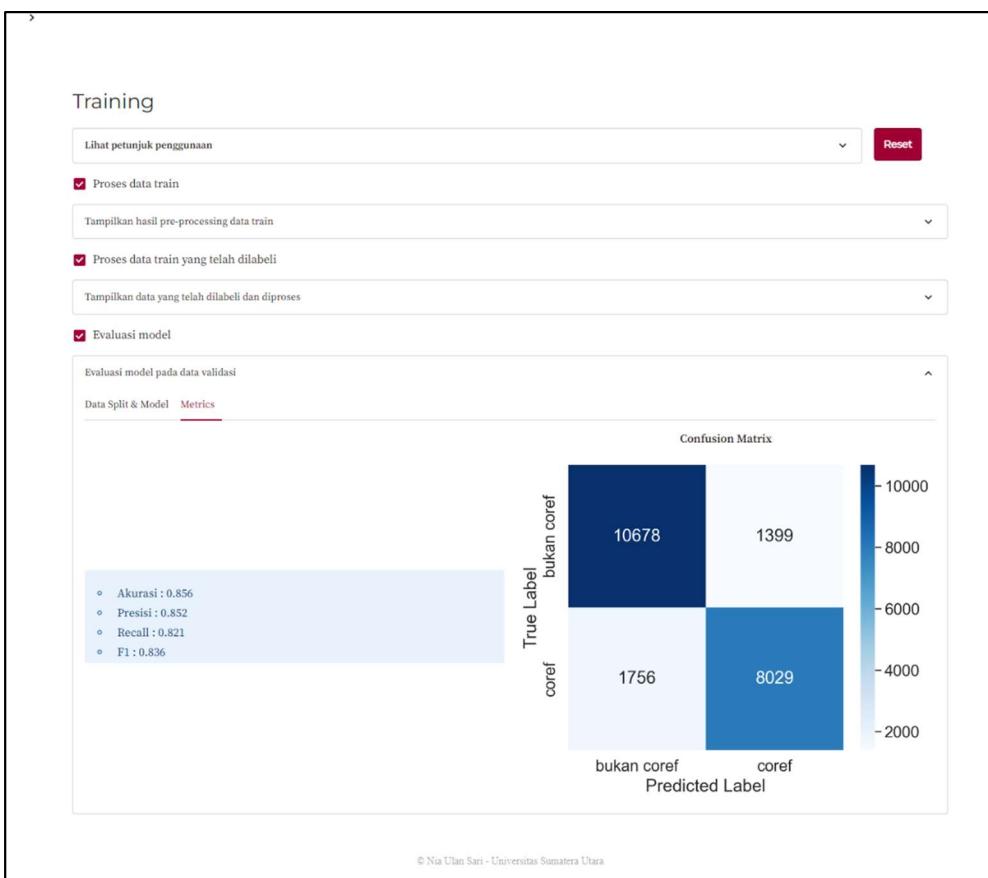


Gambar 4.5 Tampilan halaman *Training* yang menampilkan tabel hasil ekstraksi fitur

Tahap terakhir dalam proses *training* adalah evaluasi model. Centang *checkbox* “Evaluasi model” untuk menampilkan hasil evaluasi dari model pada data validasi. Berikut tampilan halaman *Training* yang menampilkan hasil evaluasi model berupa rincian *hyperparameter* model dan metriks berupa *confusion matrix* yang didapat dapat dilihat pada Gambar 4.6 dan Gambar 4.7.



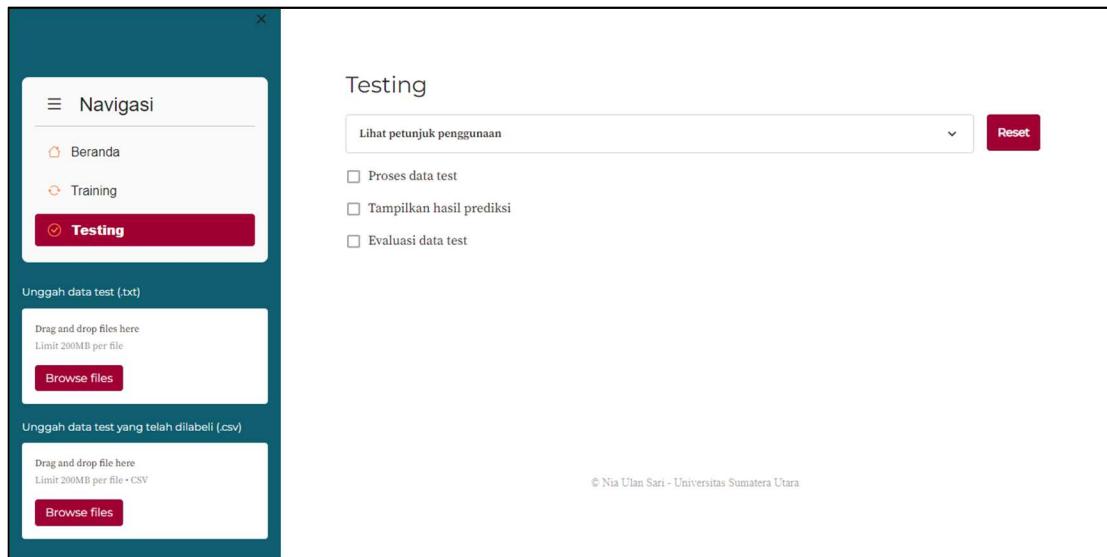
Gambar 4.6 Tampilan halaman *Training* yang menampilkan rincian *hyperparameter* model



Gambar 4.7 Tampilan halaman *Training* yang menampilkan metriks evaluasi dan *confusion matrix*

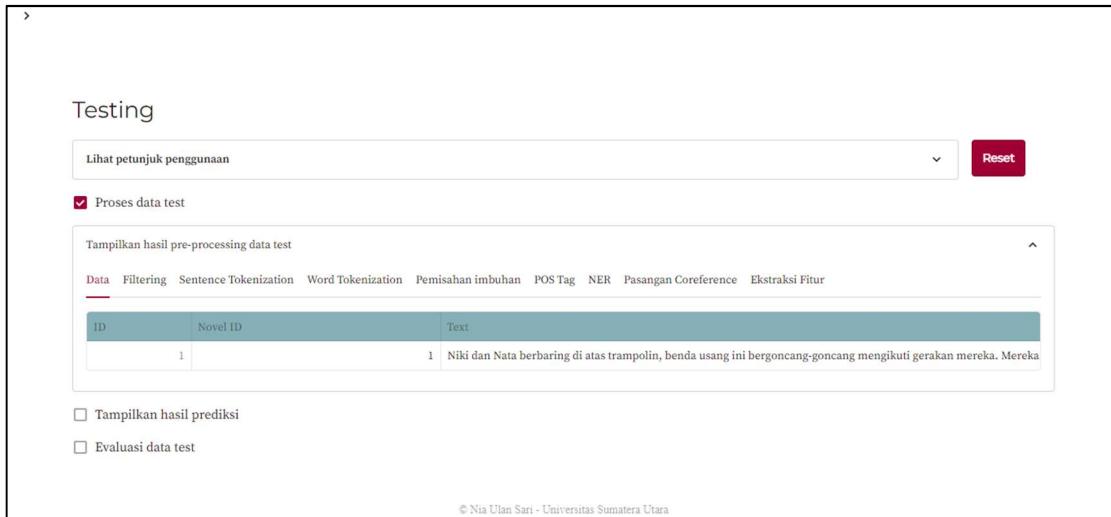
3. Tampilan halaman *Testing*

Sama dengan halaman *Training* terdapat beberapa elemen pada halaman *Testing* diantaranya adalah *expander* yang beridi kontainer penjelasan penggunaan halaman *Testing*. Kemudian tombol “Reset” untuk mengembalikan keadaan halaman ke seperti semula. Kemudian dua tombol untuk menunggah data *test* mentah dan data *test* yang sudah dilabeli. Tiga *checkbox* yang ketika diklik akan menjalankan proses *testing* dimulai dari *pre-processing* sampai hasil evaluasi data *test*. Tampilan awal halaman *Testing* dapat dilihat pada Gambar 4.8.



Gambar 4.8 Tampilan awal halaman *Testing*

Memulai proses *testing* diawali dengan menunggah data *test* dengan ekstensi *.txt* dengan mengklik tombol “*Browse files*” pertama pada *sidebar* kiri. Penelitian ini menggunakan sebanyak 10 *files* teks novel sebagai data *test*. Namun untuk proses *testing*, penulis melakukan pengujian terhadap masing-masing *file* bukan sekaligus. Setelah *file* diunggah, selanjutnya adalah centang *checkbox* “*Proses data test*” untuk menampilkan hasil *pre-processing* data mulai dari data teks sampai dengan proses NER serta pemasangan *coreference* dan ekstraksi fitur. Pengguna dapat mengklik salah satu *tab* proses untuk melihat hasil masing-masing tahapan. Tampilan halaman *Testing* yang menampilkan data teks dan hasil ekstraksi fitur dapat dilihat pada Gambar 4.9 dan Gambar 4.10.



Gambar 4.9 Tampilan halaman *Testing* yang menampilkan tabel data teks

The screenshot shows the same 'Testing' page as above, but with a larger dataset. The table now contains 10 rows of data, each corresponding to a different sentence or part of a sentence from the novel. The columns represent various linguistic features extracted from the text.

ID	novel_id_i	sentence_id_i	word_id_i	is_promoun_i	is_propernoun_i	is_plural_i	is_singular_i	is_ner_not_o_i	novel_id_j	sentence_id_j	word_id_j	is_promoun_j
1	1	1	3	0	1	0	0	1	1	1	1	1
2	1	1	15	1	0	1	0	0	0	1	1	1
3	1	1	15	1	0	1	0	0	0	1	1	3
4	1	2	17	1	0	1	0	0	0	1	1	1
5	1	2	17	1	0	1	0	0	0	1	1	3
6	1	2	17	1	0	1	0	0	0	1	1	15
7	1	4	51	1	0	1	0	0	0	1	1	1
8	1	4	51	1	0	1	0	0	0	1	1	3
9	1	4	51	1	0	1	0	0	0	1	1	15
10	1	4	51	1	0	1	0	0	0	1	2	17

At the bottom of the page, there are two checkboxes: 'Tampilkan hasil prediksi' and 'Evaluasi data test', followed by a copyright notice: '© Nia Ulan Sari - Universitas Sumatera Utara'.

Gambar 4.10 Tampilan halaman *Testing* yang menampilkan tabel hasil ekstraksi fitur

Selanjutnya pengguna dapat melihat hasil prediksi *coreference* dengan mengklik checkbox ‘‘Tampilkan hasil prediksi’’ yang menghasilkan teks dan tabel hasil prediksi *coreference* seperti pada Gambar 4.11.

The screenshot shows the 'Testing' interface with the following components:

- Header:** 'Testing' with a 'Lihat petunjuk penggunaan' link and a 'Reset' button.
- Checkboxes:**
 - Proses data test
 - Tampilkan hasil pre-processing data test
 - Tampilkan hasil prediksi
- Text Area:** 'Teks Novel' containing a paragraph of Indonesian text. Entitites and tokens are highlighted in blue, such as 'Niki' and 'Nata'. A red oval highlights the link 'Unduh data yang akan dilabeli' at the bottom of this area.
- Table:** 'Hasil prediksi' showing 10 rows of predicted coreference relations between entities 'i' and 'j'. The columns are 'ID', 'i', 'j', and 'Prediction'.

ID	i	j	Prediction
1	Nata (3)	Niki (1)	bukan coreference
2	mereka (15)	Niki (1)	coreference
3	mereka (15)	Nata (3)	coreference
4	Mereka (17)	Niki (1)	coreference
5	Mereka (17)	Nata (3)	coreference
6	Mereka (17)	mereka (15)	coreference
7	mereka (51)	Niki (1)	coreference
8	mereka (51)	Nata (3)	coreference
9	mereka (51)	mereka (15)	bukan coreference
10	mereka (51)	Mereka (17)	coreference
- Note:** A note below the table states: 'Jika ingin mengetahui evaluasi hasil prediksi, silahkan unduh data dibawah ini untuk dilabeli terlebih dahulu, kemudian unggah kembali data yang telah dilabeli tersebut pada sidebar sebelah kiri'.
- Buttons:** 'Unduh data yang akan dilabeli' (highlighted with a red oval) and 'Evaluasi data test'.

Gambar 4.11 Tampilan halaman *Testing* yang menampilkan hasil prediksi *coreference*

Tabel hasil prediksi berisi entitas dan kata ganti berlabel yang dipasangkan satu sama lain dan kemudian diidentifikasi apakah *coreference* satu sama lain atau bukan. Selanjutnya pengguna juga dapat mengetahui hasil evaluasi dari hasil prediksi dengan mengunduh data hasil tadi dengan mengklik "Unduh data yang akan dilabeli" seperti yang tertera pada Gambar 4.11(lingkar merah). Setelah data diunduh, pengguna dapat memberikan label *coreference* pada data tersebut dan lanjut ke proses selanjutnya dengan mengunggah data yang sudah dilabeli tersebut dengan mengklik tombol "*Browse files*" kedua pada sidebar kiri dan mencentang checkbox "Evaluasi data *test*" untuk menampilkan hasil evaluasi data *test* berupa tabel perbandingan prediksi dan label serta metriks berupa *confusion matrix* yang didapat dapat dilihat pada Gambar 4.12 dan Gambar 4.13.

Evaluasi hasil prediksi

ID	i	j	Label	Prediction
1	Nata (3)	Niki (1)	bukan coreference	bukan coreference
2	mereka (15)	Niki (1)	coreference	coreference
3	mereka (15)	Nata (3)	coreference	coreference
4	Mereka (17)	Niki (1)	coreference	coreference
5	Mereka (17)	Nata (3)	coreference	coreference
6	Mereka (17)	mereka (15)	coreference	coreference
7	mereka (51)	Niki (1)	coreference	coreference
8	mereka (51)	Nata (3)	coreference	coreference
9	mereka (51)	mereka (15)	coreference	bukan coreference
10	mereka (51)	Mereka (17)	coreference	coreference

© Nia Ujan Sari - Universitas Sumatera Utara

Gambar 4.12 Tampilan halaman *Testing* yang menampilkan hasil evaluasi data *test* perbandingan prediksi dan label

Confusion Matrix

		Predicted Label	
True Label	bukan coref	coref	
	bukan coref	62	13
coref	14	64	

© Nia Ujan Sari - Universitas Sumatera Utara

Gambar 4.13 Tampilan halaman *Testing* yang menampilkan metriks evaluasi dan *confusion matrix*

4.2. Pengujian Sistem

4.2.1. Implementasi model Random Forest Classifier

Metode *Random Forest Classifier* memiliki parameter yang dapat di-tuning untuk meningkatkan performansi model yang dibangun. Parameter-parameter ini dikenal dengan *hyperparameter*. *Hyperparameter* ini dapat di-tuning untuk menghasilkan model dengan performansi yang lebih baik (Clara, 2021).

Hyperparameter tuning pada *Random Forest Classifier* dilakukan pada seluruh kombinasi parameter. *Hyperparameter* metode *Random Forest Classifier* dengan menggunakan *RandomSearchCV* adalah sebagai berikut.

1. *n_estimators*, mengacu pada jumlah pohon keputusan yang akan dibangun. Pada penelitian ini, rentang *n_estimators* yang digunakan adalah 100 hingga 1000. Rentang dimulai dengan angka 100 sebagai nilai default hingga angka 1000 sebagai akhir rentang dimana penambahan jumlah pohon setelah titik ini biasanya memberikan manfaat yang semakin berkurang dalam hal peningkatan akurasi atau kinerja model secara keseluruhan. Dengan menggunakan rentang ini, diharapkan dapat menemukan keseimbangan yang optimal antara kinerja model dan efisiensi komputasi.
2. *max_features*, menentukan jumlah fitur yang dipertimbangkan untuk membagi pada setiap *node*. Pada penelitian ini, *hyperparameter max_features* yang digunakan adalah ‘auto’, ‘sqrt’, dan ‘log2’. Ketiganya merupakan nilai umum yang efektif untuk mengontrol jumlah fitur yang dipilih.
3. *max_depth*, membatasi kedalaman setiap pohon. Membatasi kedalaman dapat membantu mencegah *overfitting*. Pada penelitian ini, rentang *max_depth* yang digunakan adalah 10 hingga 100. Rentang dimulai dengan angka 10 sebagai nilai default hingga angka 100 sebagai akhir rentang dimana kedalaman pohon yang terlalu besar (lebih dari 100) dapat menyebabkan model menangkap *noise* dan detail yang tidak relevan dari data *train*, yang mengakibatkan kinerja yang buruk pada data *test*.
4. *min_samples_split*, adalah jumlah minimum sampel yang diperlukan untuk membagi *node* internal. Pada penelitian ini, rentang *min_samples_split* yang digunakan adalah 2 hingga 10. Rentang dimulai dari 2 (*default*) hingga 10 memastikan *node* tidak akan dibagi kecuali ada cukup sampel yang tersedia,

menghindari pembentukan *node* dengan sangat sedikit sampel yang bisa menjadi tidak stabil.

5. *min_samples_leaf*, adalah jumlah minimum sampel yang diperlukan untuk menjadi *leaf* pada pohon. Pada penelitian ini rentang *min_samples_leaf* yang digunakan adalah 1 hingga 10. Rentang dimulai dari 1 (*default*) hingga 10 dapat mencegah model membangun pohon yang terlalu dalam dan rumit.
6. *bootstrap*, menentukan apakah sampel *bootstrap* (sampel dengan penggantian) digunakan saat membangun pohon. Pada penelitian ini *hyperparameter bootstrap* yang digunakan adalah ‘True’ dan ‘False’. ‘True’ digunakan agar penggunaan *bagging* dapat mengurangi varians dan ‘False’ digunakan agar setiap pohon menggunakan seluruh *dataset* asli.

Dalam penelitian ini pencarian *hyperparameter* dilakukan menggunakan *RandomSearchCV* dari *library scikit-learn*. *RandomSearchCV* mencoba sejumlah kombinasi *hyperparameter* yang dipilih secara acak dari rentang *hyperparameter* yang telah ditentukan. Iterasi pencarian juga ditentukan sebanyak 100 agar *RandomSearchCV* dapat mengeksplorasi cukup banyak kombinasi *hyperparameter* yang berbeda dalam rentang yang telah ditentukan. Digunakan pula parameter *cross-validation* untuk membagi data *train* menjadi lima *subset* untuk mengevaluasi performa masing-masing kombinasi.

Untuk mendapatkan model dengan performa terbaik, penelitian ini menggunakan metrik *mean_test_score* sebagai acuan. Kombinasi *hyperparameter* dengan *mean_test_score* tertinggi akan digunakan sebagai model akhir. Performansi kombinasi *hyperparameter tuning* yang dilakukan oleh *RandomSearchCV* ditunjukkan pada Tabel 4.3.

Tabel 4.3 Performansi Hyperparameter Tuning

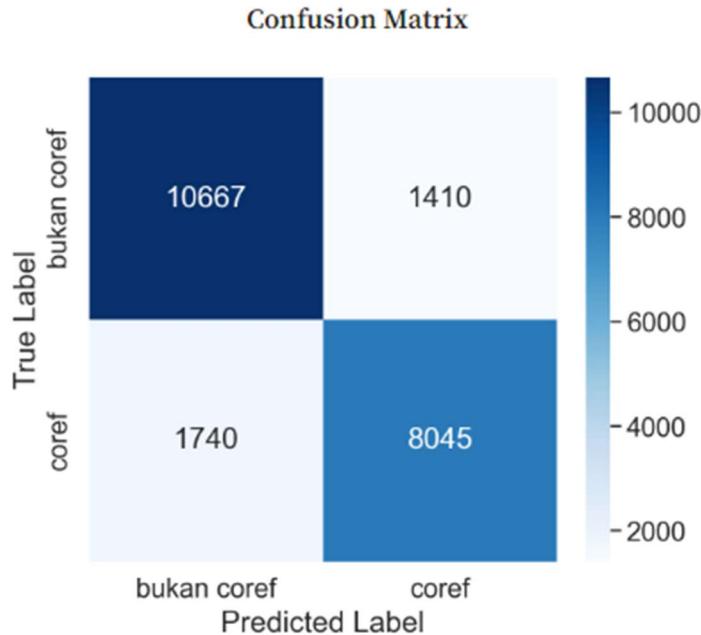
<i>iterasi</i>	<i>Hyperparameter</i>						<i>mean_test_score</i>
	<i>n_estimat</i> <i>ors</i>	<i>max_feat</i> <i>ures</i>	<i>max_dept</i> <i>h</i>	<i>min_sam</i> <i>ples_split</i>	<i>min_sam</i> <i>ples_leaf</i>	<i>bootstra</i> <i>p</i>	
1	120	auto	61	6	8	True	0,82089
2	199	log2	92	6	8	True	0,82010
3	869	log2	33	6	6	False	0,8396
4	376	sqrt	39	9	2	False	0,8466

Tabel 4.3 Performansi Hyperparameter Tuning (lanjutan)

iterasi	Hyperparameter						mean_test_score
	n_estimat ors	max_feat ures	max_dept h	min_sam ples_split	min_sam ples_leaf	bootstra p	
5	847	sqrt	85	6	6	True	0,8274
6	666	log2	58	5	3	True	0,8370
7	918	auto	73	6	3	False	0,8456
8	487	auto	30	3	7	True	0,8244
9	876	sqrt	69	3	2	True	0,8400
10	466	sqrt	62	5	4	False	0,8441
11	134	auto	80	8	8	False	0,8337
12	101	sqrt	90	5	8	False	0,8335
13	576	sqrt	15	7	4	False	0,8245
...
51	108	auto	21	9	5	False	0,8408
52	153	log2	28	9	4	True	0,8332
53	519	log2	42	9	1	False	0,847
54	609	sqrt	93	9	3	False	0,846
55	977	auto	34	9	2	True	0,8392
..
97	598	auto	53	6	2	True	0,84057
98	875	log2	44	3	1	False	0,8428
99	716	sqrt	35	4	2	True	0,8402
100	592	sqrt	77	3	5	True	0,8304

Pada Tabel 4.3, dapat disimpulkan bahwa distribusi *hyperparameter* pada iterasi ke-53 menghasilkan nilai *mean_test_score* tertinggi yaitu sebesar 0,847 diantara 99 iterasi lainnya dengan besaran nilai *n_estimators* = 519, nilai *max_features* = log2, *max_depth* = 42, *min_samples_split* = 9, *min_samples_leaf* = 1, dan *bootstrap* = False. Maka dari itu kombinasi *hyperparameter* tersebut penulis gunakan sebagai model akhir.

Model akhir yang diperoleh selanjutnya akan diimplementasikan pada data validasi untuk evaluasi kinerja model tersebut. Data validasi yang digunakan merupakan sebanyak 21862 pasangan (20%) dari total 109306 pasangan data *train* sebelum dilakukan *split*. Berikut merupakan *confusion matrix* dari implementasi model pada data validasi yang ditunjukkan menggunakan visualisasi *heatmap* pada Gambar 4.14.



Gambar 4.14 Evaluasi *Confusion Matrix* pada data validasi

Berdasarkan visualisasi *heatmap* pada Gambar 4.14 maka dapat dihitung nilai akurasi dan *f1-score* dari data validasi. Berikut perhitungannya.

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{8045 + 1066}{8045 + 10667 + 1410 + 174} = 0,856$$

$$\text{Presisi} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{8045}{8045 + 1410} = 0,851$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{F}} = \frac{8045}{8045 + 1740} = 0,822$$

$$F1 - score = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} = 2 \times \frac{0,851 \times 0,822}{0,851 + 0,822} = 0,836$$

4.2.2. Pengujian model

Setelah model selesai diuji pada data validasi dan disimpan, selanjutnya akan diuji dengan data *test* yang telah disediakan sebelumnya. Data *test* yang digunakan adalah 10 data teks novel seperti yang tertera pada Tabel 3.1 yang sudah melalui *pre-processing* sampai dengan ekstraksi fitur seperti data *train*. Fitur data teks novel “*Refrain*” ditunjukkan pada Tabel 4.4.

Tabel 4.4 Fitur data teks novel “Refrain”

Fitur (20)																			
1 1 3 0 1 0 0 1 1 1 0 1 0 0 1 0,0106 0 0 0																			
1 1 15 1 0 1 0 0 1 1 1 0 1 0 0 1 0,0741 0 0 0																			
1 1 15 1 0 1 0 0 1 1 3 0 1 0 0 1 0,635 0 0 0																			
1 2 17 1 0 1 0 0 1 1 1 0 1 0 0 1 0,847 0,769 0 0																			
1 2 17 1 0 1 0 0 1 1 3 0 1 0 0 1 0,741 0,0769 0 0																			
1 2 17 1 0 1 0 0 1 1 15 1 0 1 0 0 0 0,0106 0,0769 1 0																			
.																			
.																			
.																			
1 13 186 1 0 0 1 0 1 13 172 0 1 0 0 1 0,0741 0 0 0																			
1 13 186 1 0 0 1 0 1 13 176 0 1 0 0 1 0,529 0 0 0																			

Tiap-tiap baris fitur mewakili pasangan entitas dan kata ganti dari teks novel. Model akan memprediksi data *test* yang sudah berbentuk numerik. Hasil pengujian model terhadap data *test* teks “Refrain” ditunjukkan pada Tabel 4.5.

Tabel 4.5 Tabel hasil pengujian data *test* teks "Refrain"

No	i	j	Label	Prediksi
1	Nata (3)	Niki (1)	bukan coreference	bukan coreference
2	mereka (15)	Niki (1)	coreferene	coreference
3	mereka (15)	Nata (3)	coreference	coreference
4	Mereka (17)	Niki (1)	coreference	coreference
5	Mereka (17)	Nata (3)	coreference	coreference
6	mereka (51)	mereka (15)	coreference	coreference
7	mereka (51)	Niki (1)	coreference	coreference
8	mereka (51)	Nata (3)	coreference	coreference
9	mereka (51)	mereka (15)	coreference	coreference
10	mereka (51)	Mereka (17)	coreference	coreference
11	Mereka (81)	Niki (1)	coreference	coreference

Tabel 4.5 Tabel hasil pengujian data test teks "Refrain" (lanjutan)

No	i	j	Label	Prediksi
12	Mereka (81)	Nata (3)	coreference	coreference
13	Mereka (81)	mereka (15)	coreference	coreference
14	Mereka (81)	Mereka (17)	coreference	coreference
15	Mereka (81)	mereka (51)	coreference	coreference
16	Niki (98)	Niki (1)	coreference	coreference
17	Niki (98)	Nata (3)	bukan coreference	bukan coreference
18	Niki (98)	mereka (15)	coreference	bukan coreference
19	Niki (98)	Mereka (17)	coreference	bukan coreference
20	Niki (98)	mereka (51)	coreference	bukan coreference
21	Niki (98)	Mereka (81)	coreference	bukan coreference
22	nya (109)	Niki (1)	coreference	bukan coreference
23	nya (109)	Nata (3)	bukan coreference	bukan coreference
24	nya (109)	mereka (15)	coreference	coreference
...
144	nya (186)	nya (109)	bukan coreference	coreference
145	nya (186)	Nata (121)	bukan coreference	bukan coreference
146	nya (186)	Niki (126)	bukan coreference	bukan coreference
147	nya (186)	Dia (134)	bukan coreference	bukan coreference
148	nya (186)	Niki (139)	bukan coreference	bukan coreference
149	nya (186)	Nata (147)	bukan coreference	bukan coreference
150	nya (186)	Kak Dhanny (168)	coreference	bukan coreference
151	nya (186)	Kak Sivia (170)	bukan coreference	bukan coreference
152	nya (186)	Niki (172)	bukan coreference	bukan coreference
153	nya (186)	Nata (176)	bukan coreference	bukan coreference

Berdasarkan hasil pengujian data *test* teks "Refrain" yang terdiri dari 153 pasangan entitas dan kata ganti, didapatkan sebanyak 70 pasangan yang berlabel *coreference* diprediksi *coreference* (TP), 10 pasangan yang berlabel bukan *coreference* diprediksi *coreference* (FP), 14 pasangan yang berlabel *coreference* diprediksi bukan *coreference* (FN), dan 59 pasangan yang berlabel bukan *coreference* diprediksi bukan *coreference*

pula. Pengujian dilakukan pula pada sembilan data teks novel lainnya untuk mendapatkan hasil akhir berupa nilai metriks akurasi, presisi, *recall*, dan *f1-score* dari keseluruhan data *test*.

4.2.3. Evaluasi

Evaluasi dilakukan pada 10 data teks novel dengan menghitung nilai metriks akurasi, presisi, *recall*, dan *f1-score*. Untuk mendapatkan nilai keempat metriks tersebut diperlukan jumlah data yang salah dan benar diprediksi yang didapatkan dari *confusion matrix*. Penjabaran hasil *confusion matrix* pada tiap-tiap data *test* ditunjukkan pada tabel Tabel 4.6.

Tabel 4.6 Confusion matrix tiap data test

No	Novel	Jumlah pasangan	Confusion matrix			
			TP	FP	FN	TN
1	<i>Refrain</i>	153	70	10	14	59
2	86	741	179	64	68	430
3	Dia Tanpa Aku	253	54	11	24	164
4	Diskoneksi	820	424	24	48	324
5	Jingga Untuk Matahari	561	174	44	35	308
6	Seperti Dendam, Rindu Harus Dibayar Tuntas	91	38	4	22	27
7	Mawar Merah	528	116	40	48	324
8	Maryam	630	281	49	38	262
9	Petualangan Anak Natuna	300	156	8	28	108
10	7 Hari Menembus Waktu	861	326	66	70	399
		Total	4938	1818	320	395
						2405

Setelah *confusion matrix* tiap data *test* didapatkan, maka dapat dihitung nilai metriks akurasi, presisi, *recall*, dan *f1-score* untuk keseluruhan data *test*. Berikut perhitungan dari tiap-tiap metriks.

$$\text{Akurasi} = \frac{\sum TP + \sum TN}{\text{Juml seluruh pasangan}} \times 100 \% = \frac{1818 + 2405}{4938} \times 100 \% = 85,53\%$$

$$\text{Presisi} = \frac{\sum TP}{\sum TP + \sum FP} \times 100 \% = \frac{1818}{1818 + 320} \times 100 \% = 85\%$$

$$Recall = \frac{\sum TP}{\sum TP + \sum FN} \times 100 \% = \frac{1818}{1818 + 395} \times 100 \% = 82,2 \%$$

$$F1 - score = 2 \times \frac{\text{Presisi} \times Recall}{\text{Presisi} + Recall} \times 100 \% = 2 \times \frac{0,85 \times 0,822}{0,85 + 0,822} \times 100 \% = 83,6\%$$

Dari perhitungan di atas, nilai metriks akurasi yang diperoleh dari pengujian data *test* adalah 85,53%, presisi sebesar 85%, *recall* sebesar 82,2%, dan *f1-score* sebesar 83,6%, dan dapat disimpulkan bahwa model *Random Forest Classifier* sangat baik dalam melakukan klasifikasi pasangan *coreference* dan bukan *coreference*.

Nilai metriks *recall* lebih rendah dibandingkan dengan metriks yang lain disebabkan karena model cenderung menghasilkan lebih banyak *False Negative* (pasangan *coreference* diprediksi bukan *coreference*) daripada *False Positive* (pasangan bukan *coreference* diprediksi *coreference*) seperti terlihat pada Tabel 4.6. Nilai metriks *recall* dapat ditingkatkan lagi dengan menyeimbangkan jumlah pasangan *coreference* dan bukan *coreference* atau memperbanyak data *train* agar model bisa mengenali lebih banyak pasangan *coreference*.

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang diambil dari penelitian identifikasi *coreference resolution* untuk teks bahasa Indonesia pada teks novel menggunakan *Random forest Classifier* adalah sebagai berikut:

1. Hasil kinerja metode *Random Forest Classifier* dengan kombinasi *RandomSearchCV* menunjukkan performa yang baik dengan memperoleh nilai metriks akurasi sebesar 85,5%, presisi sebesar 85%, *recall* sebesar 82,2%, dan *f1-score* sebesar 83,6 % yang didapatkan dari pengujian 10 data *test* dengan total pasangan *coreference* dan bukan *coreference* sebanyak 4938 pasangan.
2. Nilai metriks *recall* lebih rendah dibandingkan dengan metriks lainnya disebabkan oleh banyaknya pasangan *coreference* diprediksi bukan *coreference* lebih besar dibandingkan sebaliknya.

5.2. Saran

Adapun saran dari penulis untuk pengembangan penelitian selanjutnya adalah sebagai berikut.

1. Penggunaan teks bahasa Indonesia yang lebih bervariasi, seperti teks berita dan teks bacaan lainnya.
2. Memaksimalkan kinerja POS *tagging* dan NER menggunakan metode *neural network* untuk mendapatkan hasil yang lebih baik dan akurat lagi.
3. Menyeimbangkan data *train* dengan metode tertentu agar algoritma *Random Forest Classifier* dapat lebih banyak mengenal pola kelas *coreference*.

DAFTAR PUSTAKA

- A. R. F. S. Fanoon & G. A. I. Uwanthika. 2019. Part of speech tagging for twitter conversations using conditional random fields model. *Proceedings of 2019 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, pp. 108-112.
- AlKhwiter, W. & Al-Twairesh, N. 2020. Part-of-speech tagging for arabic tweets using crf and bi-lstm. *Computer Speech & Language* 65.
- Amaral, D. O. F., Buffet, M. & Vieira, R. 2015. Comparativee analysis between notations to classify named entities using conditional random fields. *Proceedings of SymPOSium in Information and Human Language Technology*, pp. 27-31.
- Auliachman, T. & Purwarianti, A. 2019. Coreference resolution system for indonesian text with mention pair method and singleton exclusion using convolutional neural network. *Proceedings of International Conference of Advanced Informatics: Concept, Theory, and Application (ICAICTA)*, pp. 1-5.
- Badan Pengembangan dan Pembinaan Bahasa. 2016. KBBI VI Daring. (Online) <https://kbbi.kemdikbud.go.id/> (15 Desember 2023).
- Bernal, D. L., Balderas, D., Ponce, P., Rojas, M. & Molina, A. 2023. Implications of artificial intelligence algorithms in the diagnosis and treatment of motor neuron diseases – a review. *Life* 2023 13(4): 1031.
- Breiman, L. 1002. Random forests. *Machine Learning*, 45(1):5-32.
- Dinakaramani, A., Rashel, F., Luthfi, A. & Manurung, R. 2014. Designing an indonesian part of speech tagset and manually tagged indonesian corpus. *Proceedings of 2014 International Conference on Asian Language Processing (IALP)*, pp. 66-69.
- Eklund, M. 2018. Comparing feature extraction methods and efects of pre-processing methods for multilabel classification of textual data. Tesis. KTH Royal Institute Of Technology.
- Fanny, C., Waworuntu, A. & Young, A. C. 2022. Implementation of conditional random fields for named entity recognition in indonesian traditional arts digital article.

- International Journal of Multidisciplinary Research and Publications (IJMRAP).* 5(2): 51-55.
- Geeksforgeeks. 2023. Decision tree in machine learning. (Online) <https://www.geeksforgeeks.org/decision-tree-introduction-example/> (21 Januari 2024).
- Husni, S. M. & Purnamasari, K. K. Svm untuk coreference resolution bahasa indonesia yang mengandung entitas jamak. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*.
- Jeong, S.-W., Choi, M. & Choi, H.-S. 2016. Coreference resolution for korean using random forests. *KIPS Tr. Software and Data Eng* 5(11): 535-540.
- Lee, H.-Y., Surdeanu, M. & Jurafsky, D. 2017. A scaffolding approach to coreference resolution integrating statistical and rule-based models. *Natural Language Processing*, pp.1-30.
- Lourdusamy, R., & Abraham, S. 2018. A survey on text pre-processing techniques and tools. *International Journal of Computer Sciences and Engineering* 6(3):148-157.
- Ming, Kun. 2020. Chinese coreference resolution via bidirectional lstms using word and token level representations. *Proceedings of 16th International Conference on Computational Intelligence and Security (CIS)*, pp.73-76.
- Mohan, Monisha. & Nair, J. J. 2020. Coreference resolution in ambiguous pronouns using bert and svm. *Proceedings of 9th International SymPOsium on Embedded Computing and System Design (ISED)*.
- Mondal, I. 2020. Approaches to biomedical coreference resolution. *CoDS COMAD*, pp.343-344.
- Pisceldo, F., Adriani, M. & Manurung, R. 2009. Probabilistic part of speech tagging for bahasa indonesia. *3rd International MALINDO Workshop*, pp. 1-6.
- Prasetyo, V. R., Benerkah, N. & Chrisintha, V. J. 2021. Implementasi natural language processing dalam pembuatan chatbot pada program information technology universitas surabaya. *Teknika* 10(2): 114-121.
- Rahman, A., & Ng, V. 2011. Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

- Sari, A. L. 2017. Coreference resolution dengan menggunakan metode svm pada novel berbahasa indonesia. Skripsi. Universitas Komputer Indonesia.
- Sokolovska, N., Lavergne, T., Cappe, O. & Yvon, F. 2010. Efficient learning of sparse conditional random fields for supervised sequence labeling. *IEEE Journal of Selected Topics in Signal Processing* 4(6): 953-964.
- Suherik, G. J. & Purwarianti, A. 2017. Experiments on coreference resolution for indonesian language with lexical and shallow syntactic features. *Proceedings od 5th International Conference on Information and Communication technology (ICoICT)*.
- Susmitha, C. H. & Haritha, D. 2020. An ensemble technique for named entity recognition using conditional random fields and l-bfgs optimization algorithm. *International Journal of Advanced Trends in Computer Science and Engineering* 9(3): 3752-3756.
- Syaifuddin, Y. 2016. Named entity recognition for bahasa indonesia. (Online) <https://github.com/yusufsyaifudin/indonesia-ner> (15 Januari 2023).
- Tantyoko, H., Sari, D. K. & Wijaya, R. W. 2023. Prediksi potensial gempa bumi indonesia menggunakan random forest dan feature selection. *IDEALIS: InDonEsiA Journal Information System* 6(2): 83-89.
- Widhiyanti, K. & Harjoko, A. 2012. POS tagging bahasa indonesia dengan hmm dan rule based. *Jurnal Informatika*. 8(2): 151-167.



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI**
UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Jalan Universitas No. 9A Gedung A, Kampus USU Medan 20155, Telepon: (061) 821007
Laman: <http://Fasilkomti.usu.ac.id>

**KEPUTUSAN
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI
NOMOR : 2714/UN5.2.14.D/SK/SPB/2024**

**DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA**

- Membaca : Surat Permohonan Mahasiswa Fasilkom-TI USU tanggal 10 Juli 2024 perihal permohonan ujian skripsi:
- | |
|---------------------------------------------------------------------------------------------------------|
| Nama : NIA ULAN SARI |
| NIM : 171402045 |
| Program Studi : Sarjana (S-1) Teknologi Informasi |
| Judul Skripsi : Coreference Resolution Untuk Teks Bahasa Indonesia Menggunakan Random Forest Classifier |
- Memperhatikan : Bawa Mahasiswa tersebut telah memenuhi kewajiban untuk ikut dalam pelaksanaan Meja Hijau Skripsi Mahasiswa pada Program Studi Sarjana (S-1) Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara TA 2023/2024.
- Menimbang : Bawa permohonan tersebut diatas dapat disetujui dan perlu ditetapkan dengan surat keputusan
- Mengingat :
 1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.
 2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.
 3. Keputusan Rektor USU Nomor 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.
 4. Surat Keputusan Rektor USU Nomor 1876/UN5.1.R/SK/SDM/2021 tentang pengangkatan Dekan Fasilkom-TI USU Periode 2021-2026
- MEMUTUSKAN
- Menetapkan : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:
- | |
|------------------------------------------------------------------------------------------|
| Ketua : Dr. Marischa Elveny S.TI, M.Kom
NIP: 199003272017062001 |
| Sekretaris : Dedy Arisandi ST., M.Kom.
NIP: 197908312009121002 |
| Anggota Penguji : Sarah Purnamawati ST., MSc.
NIP: 198302262010122003 |
| Anggota Penguji : Dr. Romi Fadillah Rahmat, B.Comp.Sc., M.Sc.
NIP: 198603032010121004 |
| Moderator : - |
| Panitera : - |
- Kedua : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasilkom-TI USU Tahun 2024.
- Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

Tembusan :

1. Ketua Program Studi Sarjana (S-1) Teknologi Informasi
2. Yang bersangkutan
3. Arsip

Medan
Ditandatangani secara elektronik oleh:
Dekan



Maya Silvi Lydia
NIP 197401272002122001