



FINAL PROJECT - QQXXXXXX

Utilization of Named Entity Recognition for Data Extraction on Student CVs

**Aidin Ferarista Zakariyah, Alif As'ad Ramadhan, Razzan Yozha Putra,
Khairah Michiko D.W.**

Advisor

Prof. Dr. Diana Purwitasari, S.Kom., M.Sc.

Dini Adni Navastara, S.Kom, M.Sc.

Undergraduate Study Program of Artificial Intelligence Engineering

Department of Infomatics

Faculty of Intelligent Electrical and Informatics Technology

Sepuluh Nopember Institute of Technology

Surabaya

2025

ABSTRAK

Pemanfaatan Named Entity Recognition untuk Ekstraksi Data pada CV Mahasiswa

Abstrak

Proses pendaftaran untuk acara kemahasiswaan sering kali terhambat oleh inefisiensi dan potensi kesalahan akibat ekstraksi data manual dari *Curriculum Vitae* (CV). Penelitian ini mengatasi permasalahan tersebut dengan mengembangkan sistem otomatis untuk ekstraksi informasi dari dokumen CV menggunakan *Named Entity Recognition* (NER). Tujuannya adalah untuk meningkatkan efisiensi dan akurasi dalam pengolahan data pendaftar, sehingga memudahkan panitia dalam proses seleksi. Metodologi penelitian dimulai dengan pengumpulan 120 data CV, yang kemudian melalui tahap anotasi manual untuk 12 kategori entitas spesifik, termasuk informasi kontak, pendidikan, dan pengalaman. Untuk menjaga privasi, data sensitif diamankan melalui proses *masking*. Guna mengatasi keterbatasan data berlabel, penelitian ini menerapkan pendekatan *semi-supervised* dengan teknik Pseudo-Labeling untuk memperluas dataset pelatihan secara efisien. Model NER utama kemudian dilatih menggunakan *library* SpaCy pada dataset yang telah diperkaya dan divalidasi.

Hasil pengujian menunjukkan bahwa model yang dikembangkan berhasil mengekstraksi informasi dengan kinerja yang solid. Model dengan arsitektur XLMROBASE menunjukkan performa terbaik, mencapai F1-Score keseluruhan sebesar 64.54%. Model ini menunjukkan akurasi yang sangat tinggi pada entitas-entitas krusial, seperti MAI (Email) dengan F1-Score 0.941 dan NAME dengan F1-Score 0.847. Hasil ini membuktikan bahwa sistem mampu secara andal mengidentifikasi informasi penting dari CV. Sistem yang dikembangkan ini dapat diimplementasikan sebagai aplikasi mandiri atau diintegrasikan sebagai fitur dalam *Large Language Model* (LLM) untuk analisis lebih lanjut, menawarkan solusi yang fleksibel dan efektif bagi penyelenggara acara mahasiswa.

Kata Kunci: *Named Entity Recognition*, Ekstraksi Informasi, *Curriculum Vitae*, Pseudo-Labeling, SpaCy, Otomatisasi Pendaftaran.

ABSTRACT

EKSTRAKSI INFORMASI DARI DOKUMEN UNTUK PENDAFTARAN MAHASISWA

Abstract

The registration process for student events is often hindered by inefficiencies and potential errors resulting from manual data extraction from *Curriculum Vitae* (CV) documents. This research addresses this issue by developing an automated system for information extraction from CVs using *Named Entity Recognition* (NER). The objective is to enhance the efficiency and accuracy of applicant data processing, thereby simplifying the selection process for event committees. The research methodology begins with the collection of 120 CVs, which then undergo manual annotation for 12 specific entity categories, including contact information, education, and experience. To maintain privacy, sensitive data is secured through a *masking* process. To overcome the limitation of labeled data, this study implements a *semi-supervised* approach using a *Pseudo-Labelling* technique to efficiently expand the training dataset. The main NER model is then trained using the SpaCy library on the enriched and validated dataset.

Testing results demonstrate that the developed model successfully extracts information with solid performance. The model with the XLMROBASE architecture exhibited the best performance, achieving an overall F1-Score of 64.54%. This model showed very high accuracy on crucial entities, such as MAI (Email) with an F1-Score of 0.941 and NAME with an F1-Score of 0.847. These results prove that the system can reliably identify important information from CVs. The developed system can be implemented as a standalone application or integrated as a feature within a *Large Language Model* (LLM) for further analysis, offering a flexible and effective solution for student event organizers.

Keywords: *Named Entity Recognition*, Information Extraction, *Curriculum Vitae*, *Pseudo-Labelling*, SpaCy, Registration Automation.

DAFTAR ISI

ABSTRAK	1
ABSTRACT	2
DAFTAR ISI	3
DAFTAR GAMBAR	5
DAFTAR TABEL	6
DAFTAR PSEUDOCODE	7
BAB 1 PENDAHULUAN	8
1.1 Latar Belakang	8
1.2 Rumusan Masalah	9
1.3 Tujuan dan Manfaat	9
BAB 2 TINJAUAN PUSTAKA	11
2.1 Dasar Teori	11
2.1.1 Named Entity Recognition (NER)	11
2.1.2 Pseudo-Labelling	11
2.1.3 Data Masking	11
2.1.4 Transformer	12
2.1.5 Bidirectional Encoder Representation from Transformer (BERT)	12
2.1.6 XLM-RoBERTa	13
2.1.7 Decoding-enhanced BERT with Disentangled Attention (DeBERTa)	13
2.1.8 Metrik Evaluasi	13
2.1.9 Loss Function	14
2.1.10 Large Language Models (LLMs)	14
2.1.11 Masked Language Model (MLM)	15
2.1.12 Next Sentence Prediction (NSP)	15
2.1.13 Imbalanced Data	15
2.1.14 RoBERTa	15
2.1.15 F-measure	16
2.1.16 Slot Error Rate (SER)	16
BAB 3 METODOLOGI	17
3.1 Metode yang digunakan	18

3.1.1	Pengumpulan Data	18
3.1.2	Anotasi Data	18
3.1.3	CV Masking	20
3.1.4	Pelatihan Model Pseudo-Labeling	22
3.1.5	Pelatihan Model NER Spacy	23
3.1.6	Evaluasi Model NER Spacy	24
3.1.7	LLM	24
3.1.8	Deploying Model	25
3.2	Implementasi	26
3.2.1	Pengolahan Data CV dan Pseudo-Labeling	26
BAB 4	Hasil dan Pembahasan	30
4.1	Hasil Pengolahan Data	30
4.1.1	Pengumpulan dan Seleksi Data	30
4.1.2	Masking Data	30
4.1.3	Anotasi Data Manual	30
4.1.4	Hasil Pseudo-Labeling	31
4.2	Skenario Pengujian	31
4.2.1	Desain dan Alur Pengujian	31
4.2.2	Dataset dan Konfigurasi Pengujian	31
4.3	Hasil Evaluasi Model	31
4.3.1	Hasil Evaluasi Model Pseudo-Labeling	31
4.3.2	Hasil Evaluasi Model SpaCy NER	33
4.4	Perbandingan Model dan Hyperparameter	35
4.4.1	Perbandingan Model BERT dan Gliner	35
4.4.2	Pengaruh Variasi Hyperparameter	36
4.5	Integrasi dengan LLM	37
4.5.1	Tujuan Integrasi	37
4.5.2	Model yang Diintegrasikan	37
4.5.3	Skema Integrasi Sistem	38
BAB 5	Kesimpulan dan Saran	39
5.1	Kesimpulan	39
5.2	Saran	39
	DAFTAR PUSTAKA	40

DAFTAR GAMBAR

Gambar 3.1. Diagram Tahap Pelaksanaan	17
Gambar 3.2. Diagram Alur Anotasi Data	19
Gambar 3.3. Diagram Alur <i>CV Masking</i>	21
Gambar 3.4. Diagram Alur Model Pseudo-Labeling	23
Gambar 3.5. Diagram Alur Pelatihan Model NER spaCy	23
Gambar 3.6. Diagram Evaluasi Model NER spaCy	24
Gambar 3.7. Diagram Alur Kerja <i>Large Language Model</i>	25
Gambar 3.8. Diagram Alur Kerja <i>Deploying Model</i>	26
Gambar 4.1. Hasil Akhir <i>Masking Data</i>	30
Gambar 4.2. Peningkatan Performa <i>Recall</i> Selama Proses Pelatihan	32
Gambar 4.3. Peningkatan Performa <i>Precision</i> Selama Proses Pelatihan	32
Gambar 4.4. Grafik Evaluasi <i>F1-Score</i> Model Pseudo-Labeling	33
Gambar 4.5. Perkembangan Nilai <i>Recall</i> pada Model Selama Proses Pelatihan	33
Gambar 4.6. Perkembangan <i>Precision</i> pada Model NER Selama Proses Pelatihan	34
Gambar 4.7. Peningkatan <i>F1-Score</i> Sepanjang Proses Pelatihan	35

DAFTAR TABEL

Tabel 4.1 Hasil Evaluasi Model Pseudo-Labeling	31
Tabel 4.2 Perbandingan Model BERT dan Gliner	35
Tabel 4.3 Variasi Hyperparameter	36

DAFTAR PSEUDOCODE

Pseudocode 3.1. Ekstraksi Dataset Awal	27
Pseudocode 3.2. Ekstraksi Tag Unik dari Data Anotasi	28
Pseudocode 3.3. Pelatihan Model NER	29
Pseudocode 3.4. Proses Pra-Anotasi untuk Label Studio	29

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Bagi mahasiswa, Curriculum Vitae (CV) merupakan dokumen krusial yang menjadi jembatan untuk memasuki dunia profesional, baik untuk melamar program magang, beasiswa, kegiatan organisasi, maupun pekerjaan tingkat awal. Seiring dengan meningkatnya jumlah lulusan, institusi pendidikan dan para perekrut dihadapkan pada tantangan untuk memproses ribuan CV dalam waktu singkat. Oleh karena itu, sistem yang dapat melakukan ekstraksi informasi penting dari CV secara otomatis dapat menjadi alat yang sangat berharga untuk mempercepat dan mengoptimalkan proses seleksi talenta mahasiswa (Ravishankara, Reddy, & Chatterjee, 2020).

Informasi relevan yang perlu diekstraksi dari CV mahasiswa mencakup data personal, riwayat pendidikan, pengalaman organisasi, proyek akademis, dan berbagai keahlian atau kompetensi yang dimiliki (Affonso, Rossi, & de Paiva, 2020). Namun, tantangan fundamental dalam upaya otomatisasi ini terletak pada sifat dokumen CV yang tidak terstruktur dan berformat bebas (*free-form*). Mahasiswa cenderung menyajikan informasi dalam beragam format dan tata letak, sehingga mustahil untuk merancang parser sederhana—misalnya yang berbasis *regular expressions*—yang mampu mengekstraksi seluruh data secara akurat dan konsisten (Lample et al., 2016).

Kompleksitas permasalahan ini meningkat ketika sistem dituntut untuk menangani lebih dari satu bahasa, dalam kasus penelitian ini adalah Bahasa Indonesia dan Bahasa Inggris, yang menjadikan pendekatan berbasis aturan (*rule-based*) semakin tidak praktis untuk dikembangkan dan dipelihara (Pires, Schlinger, & Garrette, 2019; Purwarianti & Firdaus, 2020). Menghadapi keterbatasan tersebut, pemanfaatan teknik *machine learning* dalam kerangka *Natural Language Processing* (NLP) menjadi solusi yang paling menjanjikan. Tujuan utama dari pendekatan ini adalah mengubah dokumen CV mahasiswa yang tidak terstruktur menjadi format data yang terorganisir dan dapat dianalisis lebih lanjut (Luan, He, Ostendorf, & Hajishirzi, 2018).

Dunia NLP sendiri telah menawarkan berbagai arsitektur model canggih yang dapat diaplikasikan untuk tugas ekstraksi informasi. Pendekatan yang lebih awal dan telah menjadi dasar dalam pemrosesan sekuens adalah *Recurrent Neural Networks* (RNN), seperti *Long Short-Term Memory* (LSTM) (Hochreiter & Schmidhuber, 1997). Namun, beberapa tahun terakhir telah lahir arsitektur *transformer* yang membawa kemajuan signifikan dengan sepenuhnya mengandalkan mekanisme *attention* (Vaswani et al., 2017). Arsitektur ini memungkinkan pemrosesan data sekuensial secara paralel dan menawarkan tingkat interpretabilitas model yang lebih tinggi. Salah satu implementasi *transformer* yang paling populer adalah BERT (*Bidirectional Encoder Representations from Transformers*), yang dikenal karena kemampuannya mempelajari representasi kontekstual dua arah dan dapat di-fine-tune untuk berbagai tugas hilir dengan performa yang sangat baik (Devlin, Chang, Lee, & Toutanova, 2019).

Untuk mendapatkan pemahaman yang lebih dalam mengenai konteks informasi, pendekatan hierarkis sering kali diterapkan dengan mendefinisikan dua level informasi: section level (misalnya, Pendidikan, Pengalaman) dan item level (misalnya, nama universitas, jabatan). Section level berfungsi untuk memberikan konteks pada informasi detail yang ada di item level. Mengingat adanya beragam pilihan model, penelitian ini tidak akan terpaku pada satu arsitektur saja, melainkan akan melakukan investigasi dan perbandingan terhadap beberapa jenis model modern, seperti model berbasis RNN dan transformer, untuk menemukan arsitektur yang paling optimal bagi kasus CV mahasiswa.

Meskipun ekstraksi informasi dari CV telah menjadi subjek penelitian, banyak di antaranya berfokus pada kandidat profesional secara umum atau tidak melakukan perbandingan model secara komprehensif (Affonso et al., 2020). Selain itu, banyak pendekatan yang tidak dirancang untuk konteks multibahasa atau menggunakan model terpisah untuk setiap tugas, sehingga prosesnya tidak end-to-end. Penelitian ini mengisi celah tersebut dengan berfokus secara spesifik pada demografi mahasiswa di Indonesia dan Inggris. Salah satu kontribusi utama dari penelitian ini adalah pembangunan dataset CV mahasiswa yang telah dianotasi secara manual, karena performa model machine learning sangat bergantung pada ketersediaan data latih yang berkualitas dan relevan (Stanford NLP, 2023).

Oleh karena itu, berlandaskan pada tantangan dan peluang yang ada, penelitian ini akan melakukan perbandingan dan implementasi berbagai model NLP modern untuk membangun sistem ekstraksi informasi yang optimal dari CV mahasiswa. Hasil dari penelitian ini diharapkan dapat memberikan solusi yang efektif bagi institusi pendidikan dan industri dalam mengelola dan menemukan talenta-talenta muda secara lebih efisien dan akurat.

1.2 Rumusan Masalah

Rumusan permasalahan yang diangkat dalam Penelitian ini adalah sebagai berikut:

1. Bagaimana mengembangkan sistem yang dapat mengekstraksi informasi penting dari CV dalam format teks dan gambar?
2. Bagaimana menggunakan metode NER untuk mengidentifikasi dan mengklasifikasikan entitas informasi pada CV?
3. Bagaimana merancang skema anotasi dan membangun sebuah korpus beranotasi yang komprehensif untuk keperluan pelatihan (*training*) dan evaluasi model NER dalam konteks ekstraksi informasi CV.

1.3 Tujuan dan Manfaat

Tujuan yang ada dalam Penelitian ini adalah sebagai berikut:

1. Mengembangkan sistem otomatis untuk ekstraksi informasi penting dari dokumen CV mahasiswa.
2. Menggunakan metode NER untuk mengenali entitas seperti nama, email, nomor telepon, institusi, jurusan, pengalaman organisasi, dan keahlian.
3. Merancang skema anotasi yang sistematis serta membangun korpus CV beranotasi untuk mendukung proses pelatihan dan evaluasi model NER secara efektif.

Manfaat yang ada dalam penelitian ini adalah sebagai berikut:

1. Bagi Institusi atau Penyelenggara Event: Meningkatkan efisiensi proses pendaftaran peserta dan pengolahan data.
2. Bagi Mahasiswa: Memberikan kemudahan dalam proses pendaftaran tanpa perlu entri data berulang secara manual.
3. Bagi Pengembang Sistem: Menjadi acuan dalam pengembangan sistem ekstraksi informasi berbasis AI.

BAB 2 TINJAUAN PUSTAKA

Tinjauan pustaka ini menguraikan landasan teori dan penelitian terdahulu yang relevan dengan pengembangan sistem ekstraksi informasi dari Curriculum Vitae (CV) menggunakan kecerdasan buatan. Pembahasan difokuskan pada teknologi inti seperti, *Named Entity Recognition* (NER), serta metodologi pendukung seperti Pseudo-Labeling dan anonimisasi data yang menjadi dasar dalam penelitian ini

2.1 Dasar Teori

2.1.1 Named Entity Recognition (NER)

Named Entity Recognition (NER) adalah sebuah tugas dalam Ekstraksi Informasi yang bertujuan untuk mengidentifikasi dan mengklasifikasikan elemen-elemen informasi tertentu yang disebut *Named Entities* (NE). NER berfungsi sebagai dasar yang krusial bagi banyak bidang lain dalam Manajemen Informasi, seperti anotasi semantik, sistem tanya-jawab (*question answering*), populasi ontologi, dan penambangan opini (*opinion mining*). Istilah *Named Entity* pertama kali diperkenalkan dalam *Message Understanding Conference* (MUC) ke-6 untuk merujuk pada identifikasi entitas seperti nama orang, organisasi, dan lokasi, serta ekspresi numerik seperti waktu dan kuantitas. Meskipun demikian, definisi dari *Named Entity* itu sendiri masih menjadi perdebatan dan belum sepenuhnya jelas secara linguistik. Dari berbagai pendekatan untuk mendefinisikannya (misalnya, sebagai kata benda, penanda *rigid*, atau pengidentifikasi unik), kriteria yang paling konsisten dan dapat dipertahankan adalah berdasarkan tujuan dan domain aplikasi, di mana sebuah entitas dianggap sebagai NE karena relevan dan berguna untuk menyelesaikan masalah tertentu dalam domain tersebut (Nadeau & Sekine, 2007).

2.1.2 Pseudo-Labeling

Pseudo-Labeling adalah metode dalam pembelajaran semi-terawasi yang digunakan untuk meningkatkan performa model dengan memanfaatkan data tanpa label. Teknik ini bekerja dengan melatih model pada data berlabel, kemudian menggunakan model tersebut untuk memprediksi label pada data yang tidak berlabel. Prediksi ini disebut Pseudo-Labeling dan dianggap sebagai label tambahan yang digunakan dalam pelatihan lanjutan. Pseudo-Labeling dinilai efektif karena dapat memperluas dataset pelatihan dengan cepat tanpa biaya anotasi yang tinggi. Keberhasilan Pseudo-Labeling sangat bergantung pada keakuratan Pseudo-Labeling yang dihasilkan, karena label yang keliru dapat menurunkan performa model.

Dalam perkembangannya, Pseudo-Labeling mulai memanfaatkan *Large Language Models* (LLM) seperti ChatGPT yang mampu menghasilkan Pseudo-Labeling berkualitas tinggi melalui pendekatan *zero-shot*, *one-shot*, atau *few-shot learning*. Penggunaan LLM pada Pseudo-Labeling terbukti mampu mengurangi *noise* dan meningkatkan akurasi pada tugas-tugas yang kompleks seperti klasifikasi emosi multi-label, yang memiliki tingkat subjektivitas yang tinggi (Malik et al., 2024).

2.1.3 Data Masking

Data masking adalah sebuah proses untuk menyamarkan atau mengaburkan elemen data spesifik yang bersifat sensitif di dalam sebuah penyimpanan data. Tujuan utama dari proses ini adalah untuk memastikan bahwa informasi sensitif, seperti data pribadi pelanggan, digantikan dengan data yang realistis namun tidak nyata (*realistic but not real data*), sehingga data tersebut tidak terekspos di luar lingkungan produksi yang sah. Data masking umumnya diterapkan saat mempersiapkan lingkungan non-produksi, seperti untuk keperluan pengembangan dan pengujian (*testing*), guna menghindari risiko kebocoran data. Salah satu aspek teknis yang sangat penting dalam data masking adalah kemampuan untuk menjaga integritas referensial (*referential integrity*); oleh karena itu, algoritma yang digunakan harus dirancang agar dapat diulang (*repeatable*) sehingga nilai yang telah di-masking dapat disebarkan secara konsisten ke semua tabel terkait. Dengan menghasilkan data yang telah di-de-identified namun tetap terlihat realistis, kualitas data untuk kebutuhan pengujian, pengembangan, dan pelatihan dapat ditingkatkan secara signifikan (GK et al., 2011).

2.1.4 Transformer

Transformer adalah model *deep learning* terkemuka yang pada awalnya diusulkan sebagai arsitektur *sequence-to-sequence* untuk tugas *machine translation*. Arsitektur dasarnya terdiri dari dua komponen utama, yaitu *encoder* dan *decoder*, yang masing-masing merupakan tumpukan dari beberapa blok identik. Setiap blok pada *encoder* tersusun dari modul *multi-head self-attention* dan *position-wise feed-forward network* (FFN). Sementara itu, blok pada *decoder* memiliki struktur serupa dengan tambahan modul *cross-attention* yang berfungsi untuk menghubungkan output dari *encoder*. Karena Transformer tidak menggunakan mekanisme rekurensi atau konvolusi, ia tidak memiliki informasi urutan sekuens secara inheren, sehingga memerlukan representasi posisi tambahan (*positional encoding*) untuk memodelkan urutan token. Keunggulan utama dari mekanisme *self-attention* pada Transformer adalah kemampuannya untuk memodelkan dependensi jarak jauh (*long-range dependencies*) secara lebih efisien dan lebih mudah diparalelkan dibandingkan dengan arsitektur rekuren. Secara teoretis, Transformer memiliki sedikit asumsi mengenai bias struktural data, yang menjadikannya arsitektur yang sangat fleksibel dan universal, namun di sisi lain membuatnya rentan mengalami *overfitting* saat dilatih pada dataset berskala kecil. (Lin et al., 2022).

2.1.5 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) adalah sebuah model *deep representation-learning* yang telah menunjukkan keberhasilan besar dan mencapai hasil *state-of-the-art* dalam berbagai tugas *Natural Language Processing* (NLP). Model ini memanfaatkan arsitektur *bidirectional transformers* untuk menghasilkan representasi kata yang dikondisikan secara bersamaan pada konteks dari sisi kiri dan kanan di semua lapisan. BERT dilatih pada data mentah berskala besar dengan dua tujuan utama pra-pelatihan (*pre-training*), yaitu *Masked Language Model* (MLM) dan *Next Sentence Prediction* (NSP), yang memungkinkannya belajar representasi bahasa secara mendalam dari data tidak berlabel. Input untuk BERT terdiri dari tiga komponen: *word pieces* (sub-kata), *positions* (representasi posisi), dan *segments* (representasi segmen). Untuk tugas hilir seperti klasifikasi, sebuah token khusus ditambahkan di awal sekuens input, di mana representasi akhir dari token ini akan digunakan sebagai representasi agregat dari keseluruhan sekuens untuk diumpungkan ke

lapisan klasifikasi. Model yang telah melalui pra-pelatihan ini kemudian dapat di-*fine-tuning* dengan mudah untuk tugas-tugas spesifik.(Li et al., 2019).

2.1.6 XLM-RoBERTa

XLM-RoBERTa (XLM-R) adalah model multilingual berbasis Transformer yang dikembangkan untuk meningkatkan representasi lintas bahasa dalam skala besar. XLM-R dilatih menggunakan teknik Masked Language Modeling (MLM) pada 100 bahasa dengan data sebesar lebih dari 2 terabyte. Model ini dirancang untuk mengatasi keterbatasan model multilingual sebelumnya seperti mBERT, khususnya dalam mendukung bahasa dengan sumber daya rendah. XLM-R menunjukkan performa superior pada berbagai tugas seperti klasifikasi teks lintas bahasa (XNLI), Named Entity Recognition (NER), dan pertanyaan multibahasa (MLQA). Model ini mampu mencapai akurasi yang setara dengan model monolingual seperti RoBERTa dalam benchmark GLUE, sekaligus mempertahankan performa tinggi pada tugas-tugas multibahasa. (Conneau et al., 2020).

2.1.7 Decoding-enhanced BERT with Disentangled Attention (DeBERTa)

Decoding-enhanced BERT with Disentangled Attention (DeBERTa) adalah model transformer berbasis BERT yang dirancang untuk meningkatkan performa melalui dua mekanisme utama: *disentangled attention* dan *enhanced decoding*. Mekanisme *disentangled attention* diimplementasikan dengan pendekatan dua vektor, di mana *encoding* untuk token dan posisi dipisahkan menjadi dua vektor yang berbeda. Pemisahan ini memungkinkan lapisan *attention* untuk mempelajari dependensi dari konten dan posisi secara terpisah. Sementara itu, mekanisme *enhanced decoding* memberikan informasi posisi kata yang lebih kaya kepada model dengan menyertakan posisi relatif kata di dalam kalimat selain posisi absolutnya. Berkat kedua penyempurnaan ini, DeBERTa mampu mengungguli performa BERT dalam berbagai skenario dan telah mencapai hasil *state-of-the-art* pada banyak tugas *Natural Language Processing* (NLP), termasuk *Named Entity Recognition* (NER).(Martin et al., 2022).

2.1.8 Metrik Evaluasi

Dalam klasifikasi, khususnya pada data yang tidak seimbang, metrik seperti Precision, Recall, dan F1-score lebih informatif dibanding akurasi.

Precision mengukur proporsi prediksi positif yang benar ($TP / (TP + FP)$). Metrik ini penting saat kesalahan positif palsu perlu diminimalkan, seperti dalam deteksi spam atau diagnosis penyakit langka. Presisi tinggi umumnya membutuhkan banyak data, terutama pada kasus ketidakseimbangan kelas yang ekstrem [Juba & Le, 2019].

$$Precision = \frac{TP}{TP + FP}$$

Recall mengukur seberapa baik model menemukan semua contoh yang relevan ($TP / (TP + FN)$). Recall penting saat kelengkapan hasil lebih diutamakan daripada ketepatan, misalnya dalam pencarian literatur atau sistem deteksi dini. Pengguna sering lebih puas dengan sistem yang memiliki recall tinggi [Su, 1994].

$$Recall = \frac{TP}{TP + FN}$$

F1-score adalah rata-rata harmonik dari precision dan recall, digunakan untuk menyeimbangkan keduanya dalam satu nilai. Namun, F1-score berbasis ambang batas sehingga tidak mencerminkan distribusi kepercayaan model secara penuh, terutama pada sistem NLP skala besar. Untuk itu, pendekatan probabilistik telah diajukan sebagai alternatif [Yacouby & Axman, 2020].

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

2.1.9 Loss function

Dalam konteks *Named Entity Recognition* (NER), *loss function* (fungsi kerugian) adalah sebuah fungsi bernilai riil yang dirancang secara spesifik untuk tugas yang bersangkutan (*task-dependent*). Fungsi ini bertujuan untuk mengukur besarnya kerugian atau kesalahan yang terjadi ketika sebuah hipotesis yang dihasilkan oleh sistem tidak sesuai dengan referensi yang benar. *Loss function* untuk tugas NER dapat diturunkan secara langsung dari metrik evaluasi performa yang digunakan, seperti F-measure, *Slot Error Rate* (SER), dan fraksi slot referensi yang berhasil dikenali dengan benar. Penggunaan *loss function* yang selaras dengan kriteria evaluasi akhir ini memungkinkan proses optimisasi model yang lebih terarah pada tujuan spesifik dari tugas NER. (Goel & Byrne, 1999).

2.1.10 Large Language Models (LLMs)

Large Language Models (LLM) merupakan kategori model bahasa skala besar berbasis neural network yang dilatih pada data teks dalam jumlah masif. Model ini mampu mempelajari representasi bahasa secara statistik melalui proses pre-training, dan telah menarik perhatian luas, terutama sejak diperkenalkannya model generatif seperti ChatGPT[<https://arxiv.org/abs/2005.14165>]. Secara fundamental, LLM adalah model bahasa neural yang didasarkan pada arsitektur Transformer[attention is all you need]. Model-model ini dicirikan oleh jumlah parameternya yang masif, seringkali mencapai puluhan hingga ratusan miliar, yang dilatih pada korpora teks dalam jumlah besar, seperti "Web-scale text corpora". Skala pelatihan dan jumlah parameter yang luar biasa ini memungkinkan LLM untuk memperoleh kemampuan pemahaman dan generasi bahasa tujuan umum yang sangat kuat.

Fungsi utama dari LLM adalah memprediksi token (kata atau subkata) berikutnya dalam suatu urutan. Kemampuan prediktif ini memungkinkan model untuk secara efektif memahami konteks dan menghasilkan teks yang koheren serta relevan secara kontekstual.

2.1.11 *Masked Language Model (MLM)*

Masked Language Model (MLM) adalah salah satu dari dua tujuan pra-pelatihan (*pre-training objective*) yang digunakan oleh model BERT, selain *Next Sentence Prediction (NSP)*. Tujuan dari penggunaan MLM adalah agar model BERT dapat belajar representasi bahasa secara mendalam dari data mentah atau tidak berlabel dalam skala besar. (Li et al., 2019).

2.1.12 *Next Sentence Prediction (NSP)*

Next Sentence Prediction (NSP) adalah tugas yang bertujuan untuk memprediksi apakah dua kalimat disusun secara berurutan dalam sebuah dokumen. NSP awalnya diperkenalkan sebagai bagian dari pre-training pada model BERT untuk membantu pemahaman hubungan antar kalimat. Dalam pengembangannya, NSP terbukti efektif untuk meningkatkan kemampuan model dalam memahami konteks di tingkat kalimat dan dokumen.

NSP-BERT menghidupkan kembali konsep NSP dengan pendekatan prompt-based learning, di mana model dilatih menggunakan template kalimat yang dirancang khusus untuk menilai keterkaitan antar kalimat. Berbeda dengan metode masked language modeling (MLM) yang fokus pada level token, NSP-BERT bekerja pada level kalimat dan mampu menangani berbagai tugas seperti klasifikasi kalimat tunggal, sentence-pair, dan entity linking. Model ini terbukti efektif dalam skenario zero-shot dan few-shot learning, bahkan mampu menyaingi beberapa metode few-shot modern. (Sun, Zheng, Hao, & Qiu, 2021)

2.1.13 *Imbalanced Data*

Imbalanced Data adalah kondisi pada dataset di mana distribusi antar kelas tidak seimbang, yaitu jumlah data pada satu kelas jauh lebih banyak dibandingkan kelas lainnya. Situasi ini sering terjadi pada permasalahan dunia nyata seperti deteksi penipuan, diagnosis penyakit langka, atau sistem keamanan, di mana data dari kelas minoritas sulit diperoleh. Imbalanced data dapat menyebabkan model pembelajaran mesin menjadi bias terhadap kelas mayoritas, sehingga menghasilkan akurasi yang tinggi secara keseluruhan, namun dengan performa yang buruk dalam mendeteksi kelas minoritas.

Beberapa pendekatan umum untuk mengatasi masalah *imbalanced data* antara lain adalah teknik pre-processing seperti under-sampling, yaitu mengurangi jumlah sampel pada kelas mayoritas, dan over-sampling, yaitu menambah sampel pada kelas minoritas baik secara acak maupun dengan metode sintesis seperti SMOTE. Selain itu, pendekatan cost-sensitive learning juga banyak digunakan, yaitu memberikan bobot kesalahan yang lebih besar pada kelas minoritas agar model lebih sensitif dalam mendeteksi data dari kelas tersebut. Pemilihan metode penanganan imbalance yang tepat sangat bergantung pada jenis model yang digunakan dan karakteristik dataset yang dihadapi. (Yu & Zhou, 2021).

2.1.14 RoBERTa

RoBERTa (Robustly Optimized BERT Pretraining Approach) adalah model pengembangan dari BERT yang dirancang untuk meningkatkan kinerja pre-training melalui optimasi teknik pelatihan. RoBERTa menghilangkan komponen Next Sentence Prediction (NSP) yang dianggap tidak memberikan kontribusi signifikan pada performa, serta menggunakan teknik dynamic masking, di mana token yang disembunyikan selama pelatihan diacak setiap kali data diulang. Selain itu, RoBERTa dilatih dengan batch yang lebih besar, waktu pelatihan yang lebih lama, dan memanfaatkan kumpulan data yang lebih besar seperti CC-News, OpenWebText, dan Stories, sehingga total data mencapai sekitar 160 GB.

Melalui optimasi tersebut, RoBERTa berhasil mencapai performa yang lebih tinggi dibandingkan model BERT pada berbagai benchmark seperti GLUE, SQuAD, dan RACE. Keunggulan RoBERTa terletak pada kemampuannya mempelajari representasi bahasa yang lebih efektif dan generalisasi yang lebih baik.(Liu et al., 2019).

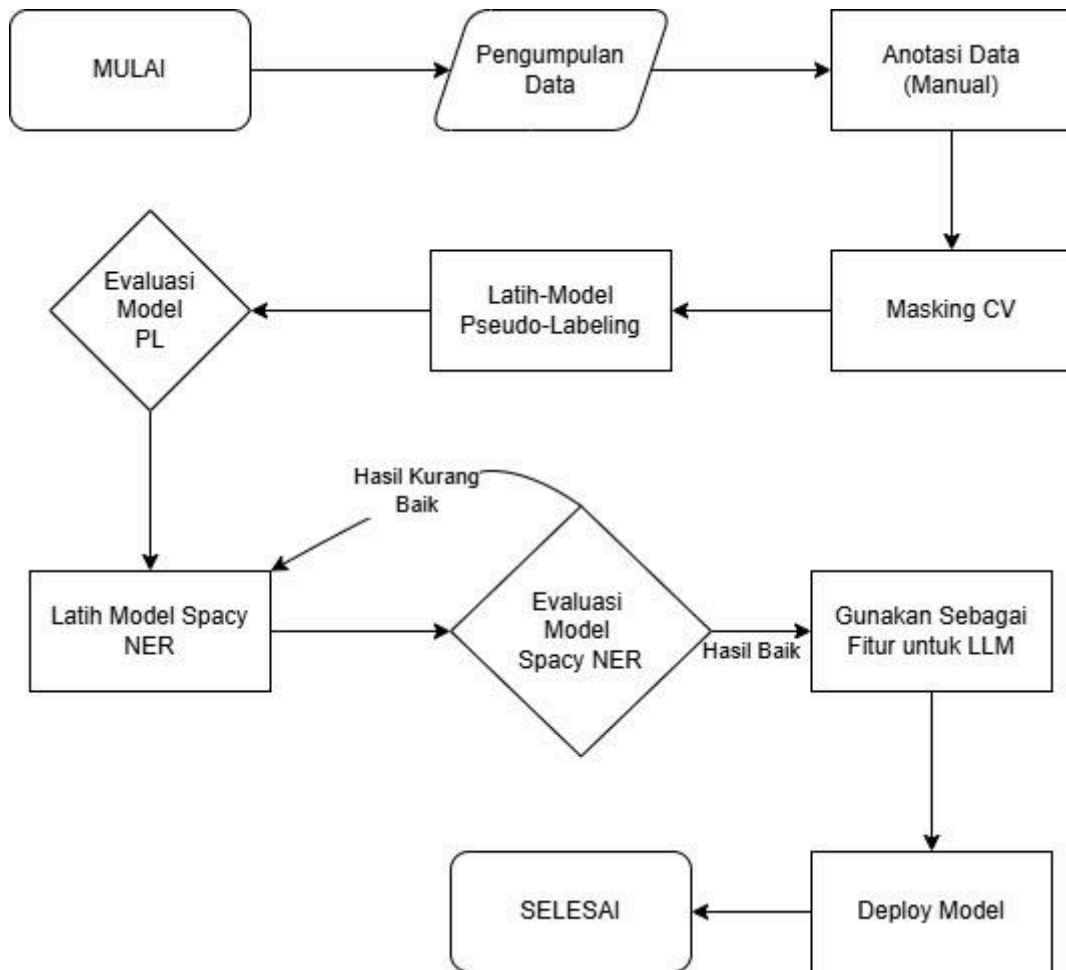
2.1.15 F-measure

F-measure, yang sering disebut sebagai F1-score ketika bobotnya seimbang, adalah metrik evaluasi yang didefinisikan sebagai rata-rata harmonik (*harmonic mean*) dari *Precision* (P) dan *Recall* (R). Penggunaan rata-rata harmonik ini lebih intuitif dibandingkan rata-rata aritmatika karena akan menghasilkan skor yang rendah jika salah satu dari nilai *Precision* atau *Recall* sangat rendah, sehingga memberikan gambaran performa yang lebih seimbang dan realistis. Secara lebih umum, F-measure memiliki formula yang dapat disesuaikan melalui parameter β , yang berfungsi untuk mengontrol keseimbangan antara *Precision* dan *Recall*. Nilai $\beta=1$ memberikan bobot yang sama (menghasilkan F1-score), nilai $\beta>1$ akan lebih menekankan *Recall*, sementara nilai $\beta<1$ akan lebih menekankan *Precision*. Asal-usul formula ini sendiri berakar dari fungsi E (efektivitas) yang diusulkan oleh van Rijsbergen.(Sasaki, 2007).

2.1.16 Slot Error Rate (SER)

Slot Error Rate (SER) adalah metrik yang digunakan untuk mengukur kesalahan sistem dalam mengenali dan mengisi slot informasi. SER dihitung berdasarkan jumlah kesalahan berupa substitution (slot salah), deletion (slot hilang), dan insertion (slot tambahan), yang dibandingkan dengan jumlah slot yang seharusnya ada. SER memberikan gambaran langsung tentang total kesalahan sistem dan sering digunakan dalam evaluasi sistem dialog dan ekstraksi informasi. [Makhoul, J., Kubala, F., Schwartz, R., & Weischedel, R. "Performance Measures for Information Extraction." BBN Technologies & GTE Corp. (1999).]

BAB 3 METODOLOGI



Gambar 3.1 Diagram tahap pelaksanaan

Penelitian ini diawali dengan anotasi manual pada korpus data berisi 120 *Curriculum Vitae* (CV). Sebanyak 12 kategori entitas spesifik didefinisikan untuk mengekstrak informasi kontak (nama, email, telepon), pengalaman profesional (organisasi, jabatan, durasi), latar belakang akademis (institusi, gelar), kualifikasi (keahlian, bahasa), serta informasi tanggal. Setelah anotasi, data CV dianonimkan melalui proses *masking* untuk melindungi informasi pribadi (PII).

Selanjutnya, untuk memperluas dataset, diterapkan metode *semi-supervised* melalui Pseudo-Labeling. Sebuah model awal yang dilatih pada data manual digunakan untuk memprediksi label pada data sisa secara iteratif, guna memastikan kualitas anotasi dan meminimalkan propagasi error. Dataset yang telah diperluas ini kemudian digunakan untuk melatih model *Named Entity Recognition* (NER) utama menggunakan Spacy, yang kinerjanya dievaluasi secara ketat dengan metrik *precision*, *recall*, dan *F1-score*.

Model NER yang telah tervalidasi dan mencapai kinerja solid dirancang untuk dua skenario penerapan. Model ini dapat di-*deploy* sebagai aplikasi mandiri untuk ekstraksi CV

otomatis, atau diintegrasikan sebagai fitur pada *Large Language Model* (LLM) untuk meningkatkan kemampuan ekstraksi data terstruktur. Pendekatan ganda ini menawarkan fleksibilitas penerapan model yang luas.

3.1. Metode yang digunakan

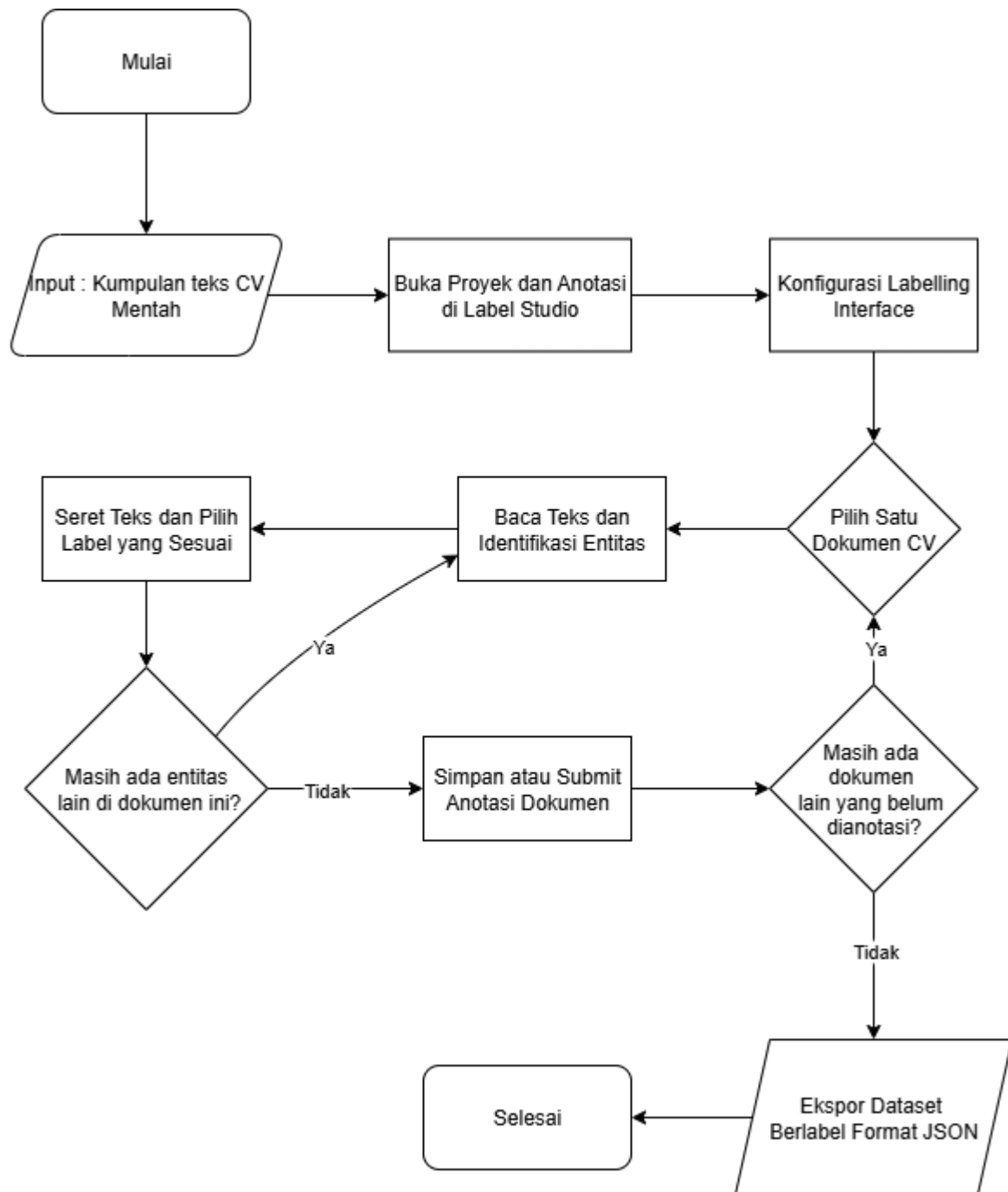
3.1.1. Pengumpulan data

Dataset yang digunakan merupakan data yang diambil dari 2 kegiatan mahasiswa yang berbeda dan 1 data hasil scraping manual sebuah perusahaan dengan total keseluruhan data berjumlah 120 data. Dataset ini berisi kumpulan Curriculum Vitae (CV) pelamar kegiatan mahasiswa dan perusahaan.

3.1.2. Anotasi Data

Tahap krusial dalam alur kerja *supervised learning* untuk *Named Entity Recognition* (NER) adalah pembuatan dataset berlabel (*annotated dataset*) berkualitas tinggi. Dataset ini berfungsi sebagai *ground truth* atau "kunci jawaban" yang akan digunakan untuk melatih dan mengevaluasi kinerja model. Mengingat belum tersedianya dataset CV berbahasa Indonesia yang beranotasi publik dengan skema yang spesifik, maka diperlukan proses anotasi. Anotasi data dilakukan secara manual dengan bantuan aplikasi web interaktif Label Studio.

1. **Pengaturan Proyek:** Sebuah proyek baru dibuat di dalam dasbor Label Studio. Pada tahap ini, *template* antarmuka pelabelan (*labeling interface*) diatur secara spesifik untuk tugas NER pada teks. Skema anotasi yang telah didefinisikan sebelumnya, yang terdiri dari 12 label entitas (NAME, EMAIL, PHONE, EDU, DEGREE, ORG, ROLE, DURATION, DAT, SKILL, LANG, dan ACH), dikonfigurasi menggunakan sintaks XML di dalam pengaturan antarmuka.
2. **Impor Data:** Seluruh data CV mentah (dalam format teks) yang telah dikumpulkan kemudian diimpor ke dalam proyek Label Studio untuk disiapkan sebagai *task* (tugas) anotasi.
3. **Proses Pelabelan:** Anotator manusia secara manual membaca setiap dokumen CV satu per satu. Ketika sebuah entitas yang relevan ditemukan, anotator akan menyorot (*highlight*) potongan teks tersebut dan memilih label yang sesuai dari panel label yang tersedia. Proses ini diulangi hingga seluruh entitas yang dapat diidentifikasi dalam satu dokumen selesai ditandai.
4. **Ekspor Hasil:** Setelah sejumlah data selesai dianotasi, hasilnya diekspor menjadi satu file JSON. File ini memiliki struktur standar yang memuat teks asli dari setiap CV beserta daftar entities yang berisi posisi karakter awal (start), posisi karakter akhir (end), dan nama label untuk setiap entitas yang telah ditandai. Dataset berformat inilah yang kemudian menjadi input utama untuk tahap pra-pemrosesan dan pelatihan model.



Gambar 3.2 Diagram alur anotasi data

Label	Deskripsi
NAME	Nama lengkap atau nama panggilan dari individu atau kandidat.
MAIL	Alamat email kandidat yang valid.
PHONE	Nomor kontak kandidat.
EDU	Nama resmi dari institusi pendidikan,

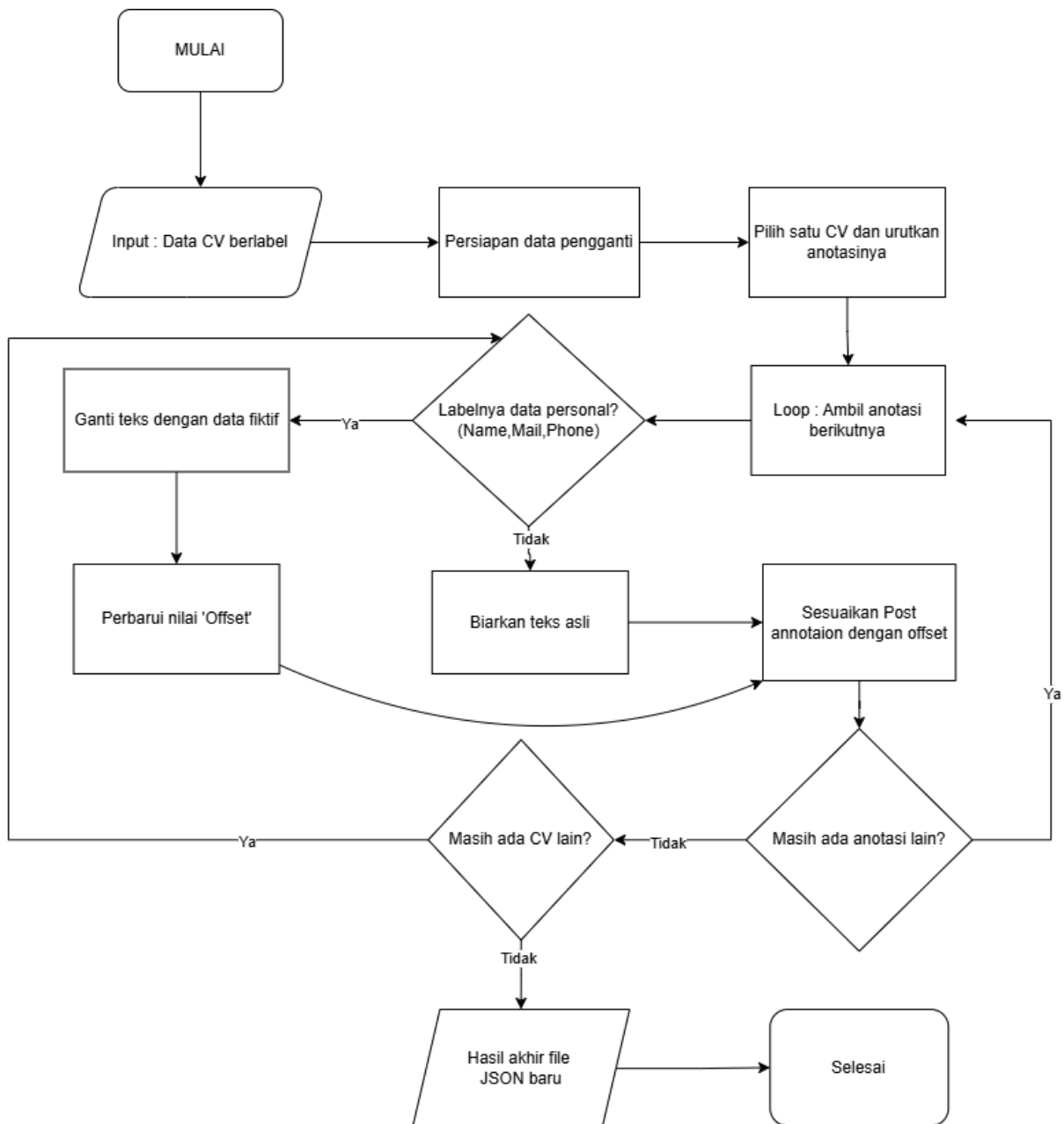
	seperti universitas atau sekolah menengah (SMA/SMK).
DEGREE	Jurusan, program studi, atau gelar akademik yang diambil. Termasuk juga jenjang studi atau peminatan.
ORG	Nama resmi dari organisasi, perusahaan, unit kegiatan mahasiswa (UKM), kepanitiaan, acara/event, kompetisi, atau workshop.
ROLE	Posisi atau jabatan seseorang dalam sebuah organisasi, kepanitiaan, atau pengalaman kerja.
DURATION	Periode atau rentang waktu yang memiliki titik awal dan akhir yang jelas, baik itu antar tanggal atau antar tahun.
DAT	Titik waktu tunggal yang spesifik, seperti tahun, bulan, dan tahun, atau tanggal lengkap.
SKILL	Kemampuan teknis (<i>hard skill</i>) atau non-teknis (<i>soft skill</i>) yang dimiliki
LANG	Bahasa yang dikuasai oleh kandidat.
ACH	Penghargaan, peringkat dalam kompetisi, beasiswa, atau sertifikasi yang telah diraih.

3.1.3. CV Masking

tahap selanjutnya dalam persiapan data adalah melakukan *masking* atau anonimisasi. Tujuan utama dari tahap ini adalah untuk melindungi Informasi Identitas Pribadi (*Personally Identifiable Information* - PII) yang terkandung dalam setiap dokumen CV. Proses ini krusial untuk menjaga kerahasiaan data pendaftar dan memenuhi standar etika penelitian saat data akan diolah lebih lanjut atau dibagikan.

Proses masking ini secara spesifik menargetkan entitas-entitas yang paling sensitif. Berdasarkan skema anotasi yang telah didefinisikan, tiga jenis entitas berikut diidentifikasi sebagai PII dan menjadi target untuk anonimisasi:

- NAME (Nama Lengkap Kandidat)
- MAIL (Alamat Email)
- PHONE (Nomor Telepon)



Gambar 3.3 Diagram alur CV Masking

Untuk mengimplementasikan proses ini, sebuah skrip Python dikembangkan yang secara sistematis membaca file JSON hasil ekspor dari Label Studio. Alur kerja tersebut adalah sebagai berikut:

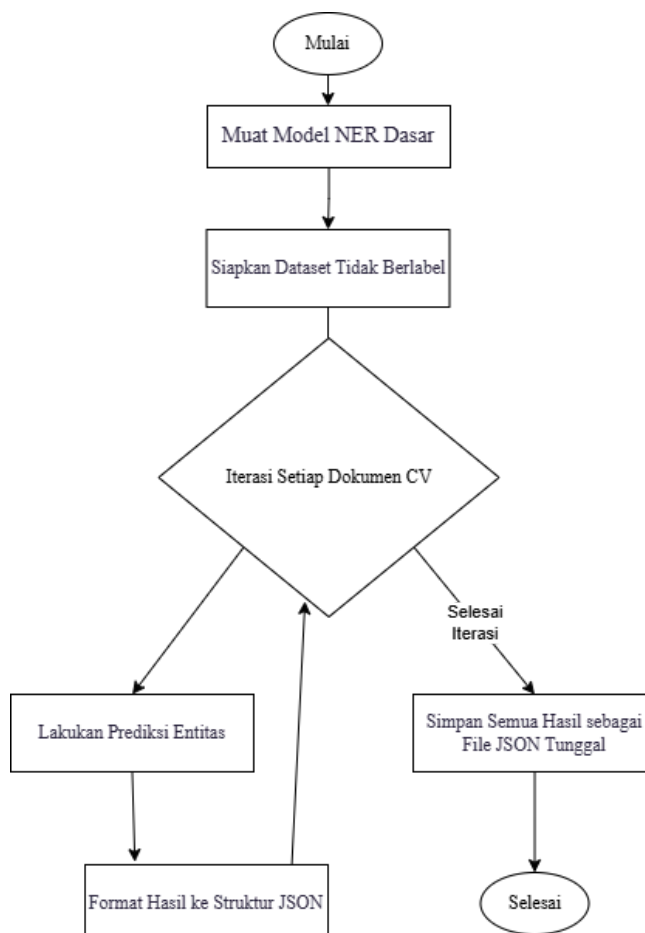
1. Persiapan Data Pengganti: Skrip terlebih dahulu menyiapkan sebuah daftar berisi 100 nama fiktif khas Indonesia. Selain itu, terdapat fungsi untuk menghasilkan alamat email dan nomor telepon fiktif secara acak dengan format yang realistis.
2. Iterasi dan Penggantian Teks: Skrip mengiterasi setiap entitas yang telah dianotasi dalam satu CV.
 - a. Jika sebuah entitas memiliki label NAME, MAIL, atau PHONE, teks aslinya akan diganti dengan data fiktif yang sesuai dari generator yang telah disiapkan.

- b. Entitas dengan label lain (seperti SKILL, ORG, EDU, dll.) tidak diubah dan teksnya dibiarkan seperti aslinya.
3. Penyesuaian Posisi Anotasi (Offset Adjustment): Ini adalah bagian paling kritis dari proses masking. Ketika sebuah teks asli (misalnya, "Emha Maulana Firdaus") diganti dengan teks fiktif (misalnya, "Citra Dewi"), panjang keseluruhan dari dokumen CV akan berubah. Perubahan panjang ini akan menyebabkan posisi karakter (*character offset*) dari semua anotasi berikutnya menjadi tidak valid. Untuk mengatasi ini, skrip secara cerdas melakukan:
 - a. Mengurutkan semua anotasi berdasarkan posisi awalnya.
 - b. Menyimpan sebuah variabel offset yang terus melacak total perubahan panjang teks.
 - c. Setelah setiap penggantian teks, nilai start dan end dari semua anotasi berikutnya diperbarui dengan menambahkan nilai offset saat ini.

Hasil akhir dari tahap ini adalah sebuah file JSON baru.

3.1.4. Pelatihan Model Pseudo-Labeling

Untuk mempercepat proses anotasi pada dataset besar yang belum berlabel, penelitian ini mengadopsi strategi Pseudo-Labeling. Tahap pertama dari strategi ini, yaitu pra-anotasi, bertujuan untuk menghasilkan label awal secara otomatis. Alur kerja dari proses pra-anotasi ini diilustrasikan pada Gambar 3.



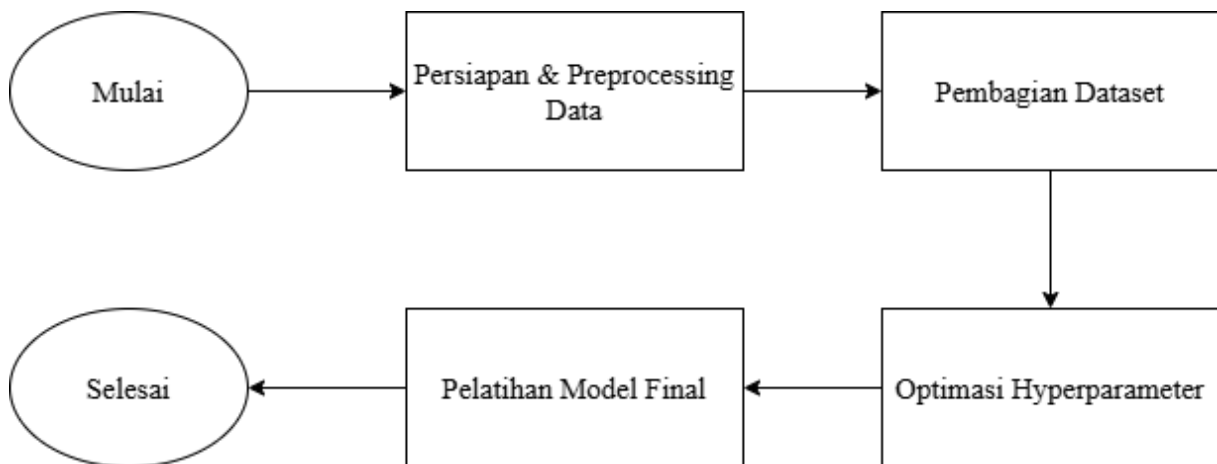
Gambar 3.4 Diagram alur model Pseudo-Labeling

Proses pra-anotasi atau Pseudo-Labeling bertujuan untuk memberikan label awal secara otomatis pada dataset yang tidak berlabel menggunakan model yang telah dilatih sebelumnya. Alur kerja ini diawali dengan memuat model NER dasar yang telah dilatih pada data manual, beserta tokenizer dan pemetaan label yang sesuai dari direktori penyimpanannya. Selanjutnya, dataset tidak berlabel, yang dalam penelitian ini berupa kumpulan dokumen CV dalam format teks, disiapkan dengan mengekstraknya dari sebuah file kompresi.

Sistem kemudian melakukan iterasi pada setiap dokumen CV tidak berlabel yang tersedia. Untuk setiap dokumen, model melakukan prediksi entitas untuk mengidentifikasi dan melabeli informasi relevan secara otomatis. Setelah prediksi selesai, hasilnya diformat ke dalam struktur JSON yang kompatibel dengan platform anotasi Label Studio. Setelah semua dokumen selesai diproses, keseluruhan hasil prediksi yang telah diformat disimpan sebagai satu file JSON tunggal. Kualitas dari hasil Pseudo-Labeling ini kemudian dapat dievaluasi secara kualitatif dengan memeriksa langsung isi dari file JSON yang dihasilkan. File inilah yang nantinya dapat diimpor ke platform anotasi untuk divalidasi dan dikoreksi oleh anotator manusia, sebelum digunakan kembali untuk melatih model pada tahap selanjutnya.

3.1.5. Pelatihan model NER Spacy

Metodologi penelitian yang digunakan dalam pengembangan model Named Entity Recognition (NER) ini terdiri dari beberapa tahapan utama. Alur kerja dari setiap tahapan, mulai dari persiapan data hingga pelatihan model, disajikan secara visual pada Gambar 4.



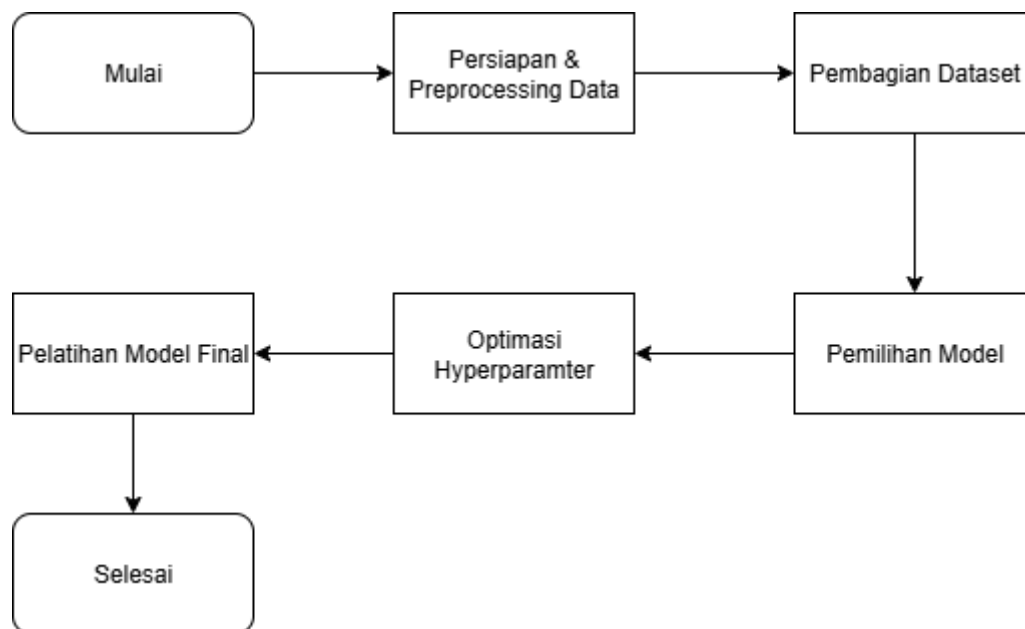
Gambar 3.5 Diagram alur pelatihan model *NER Spacy*

Alur kerja penelitian ini diawali dengan tahap persiapan dan preprocessing data. Pada tahap ini, dataset yang bersumber dari file JSON hasil anotasi manual dibersihkan terlebih dahulu untuk memastikan kualitasnya. Proses pembersihan ini mencakup penghapusan spasi tambahan pada setiap entitas yang telah dilabeli. Setelah bersih, data dikonversi ke dalam format biner *.spacy* (DocBin), yang merupakan format efisien untuk diproses oleh *pipeline* pelatihan SpaCy.

Selanjutnya, dilakukan pembagian dataset menggunakan library Scikit-learn, di mana data secara acak dibagi menjadi dua set: 80% sebagai data latih dan 20% sebagai data uji. Tahap berikutnya adalah optimasi hyperparameter, yang bertujuan untuk menemukan konfigurasi model terbaik. Proses ini diotomatisasi menggunakan platform Weights & Biases, yang menjalankan serangkaian eksperimen untuk menguji berbagai kombinasi parameter seperti learning rate, dropout, dan ukuran batch. Konfigurasi yang menghasilkan F-score entitas (*ents_f*) tertinggi dipilih untuk digunakan pada tahap akhir. Terakhir, dilakukan pelatihan model final menggunakan seluruh data latih dan konfigurasi hyperparameter optimal yang telah ditemukan. Proses pelatihan ini dijalankan dengan akselerasi GPU untuk mempercepat waktu komputasi hingga model siap untuk digunakan.

3.1.6. Evaluasi model NER Spacy

Setelah model berhasil dilatih, tahap selanjutnya adalah melakukan evaluasi untuk mengukur performa dan kapabilitasnya. Alur kerja dari proses evaluasi ini diilustrasikan secara rinci pada Gambar 5.

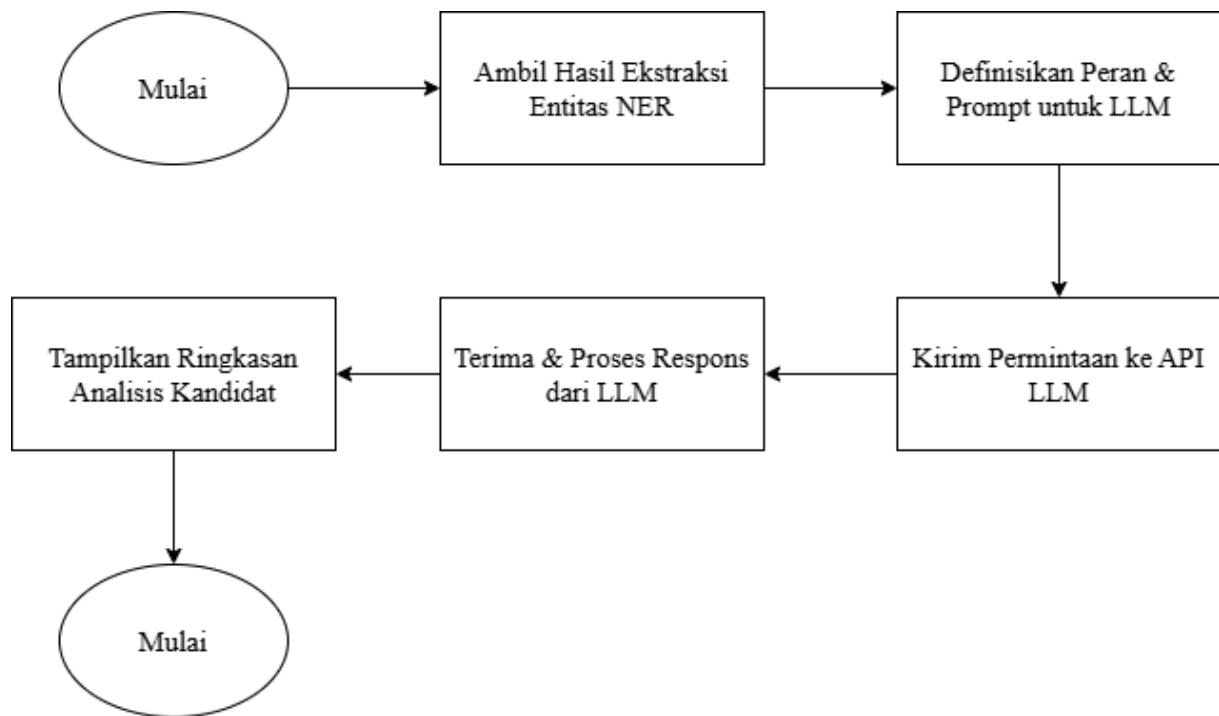


Gambar 3.6 Diagram evaluasi model *NER Spacy*

Setelah proses pelatihan selesai, tahap selanjutnya adalah evaluasi model. Tahap ini bertujuan untuk menguji kemampuan dan performa model NER yang telah dilatih pada data yang sepenuhnya baru. Proses ini diawali dengan memuat model terlatih dari direktori penyimpanannya. Selanjutnya, sebuah dokumen baru yang tidak termasuk dalam data pelatihan disiapkan sebagai input. Teks dari dokumen tersebut diekstraksi secara penuh untuk dijadikan data uji kualitatif, kemudian dimasukkan ke dalam alur kerja (pipeline) model untuk melakukan prediksi entitas. Akhirnya, hasil prediksi dari model diekstrak dan ditampilkan, di mana setiap entitas yang terdeteksi beserta labelnya disajikan untuk dianalisis guna memvalidasi kemampuan model dalam mengenali informasi secara praktis..

3.1.7. LLM

Setelah informasi diekstraksi oleh model NER, dilakukan tahap analisis lebih lanjut menggunakan LLM. Alur kerja untuk tahap analisis berbasis LLM ini diilustrasikan pada Gambar 6.

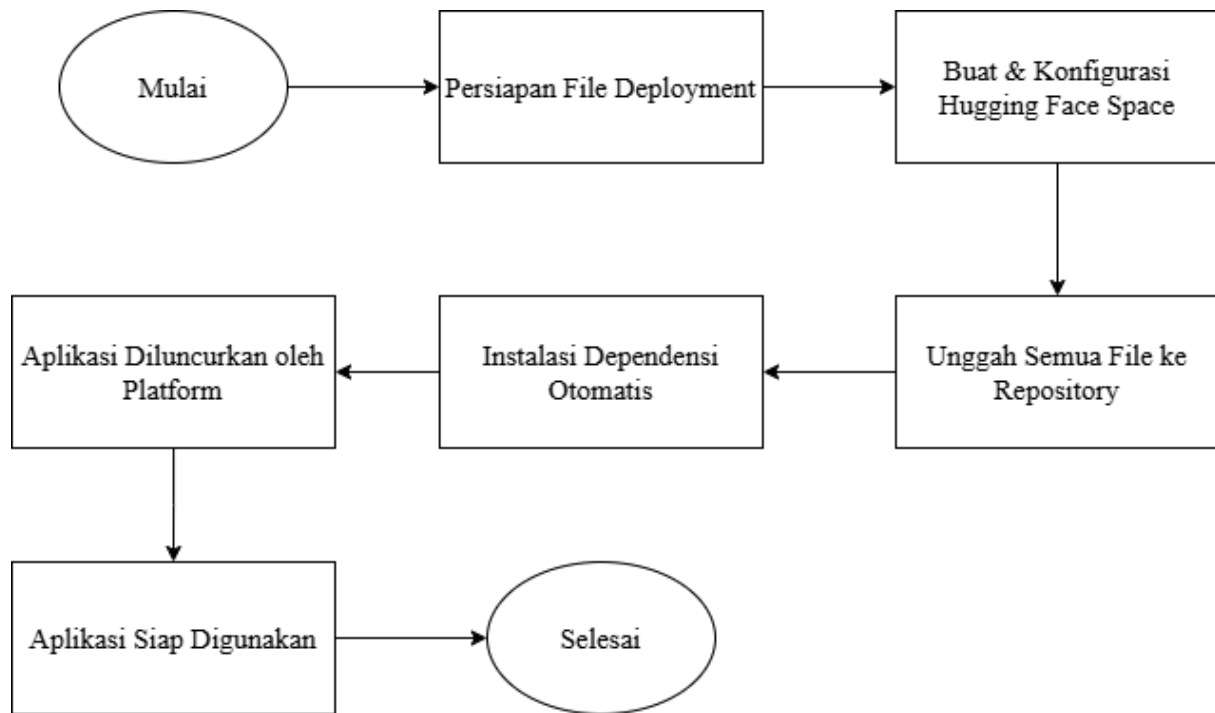


Gambar 3.7 Diagram alur kerja *Large Language Model*

Proses analisis lanjutan ini diawali dengan mengambil hasil ekstraksi entitas yang telah diprediksi oleh model NER sebagai input utama. Selanjutnya, dilakukan tahap rekayasa prompt (prompt engineering), di mana sebuah peran dan instruksi (prompt) yang detail didefinisikan untuk LLM. Prompt ini menginstruksikan LLM untuk bertindak sebagai seorang *asisten recruiter* senior yang analitis, lengkap dengan kriteria evaluasi dan format output spesifik yang harus diikuti. Setelah prompt siap, sistem mengirimkan permintaan ke API LLM, di mana hasil ekstraksi NER dan posisi pekerjaan yang relevan digabungkan ke dalam permintaan tersebut. Sistem kemudian menerima dan memproses respons yang dihasilkan oleh LLM, dengan mengekstrak konten teks dari struktur data JSON yang diterima. Akhirnya, ringkasan analisis kandidat yang telah dibuat oleh LLM sesuai format yang diinstruksikan ditampilkan sebagai output akhir dari keseluruhan sistem.

3.1.8. Deploying Model

Untuk membuat model NER yang telah dikembangkan dapat diakses dan digunakan oleh publik, dilakukan tahap deployment. Proses ini mengubah model menjadi sebuah aplikasi web interaktif menggunakan platform Hugging Face Spaces. Alur kerja dari tahap deployment ini diilustrasikan pada Gambar 7.



Gambar 3.8 Diagram alur kerja *Deploying Model*

Untuk mendeploy model agar dapat diakses secara publik, penelitian ini memanfaatkan platform Hugging Face Spaces. Tahap persiapan file deployment menjadi langkah awal, di mana beberapa komponen kunci disiapkan. Komponen tersebut meliputi skrip aplikasi utama (*app.py*) yang mengatur logika antarmuka dan pemrosesan input, file *requirements.txt* yang berisi daftar semua dependensi library Python, serta paket model (*.tar.gz*) yang telah dilatih sebelumnya.

Setelah semua file siap, langkah selanjutnya adalah membuat dan mengkonfigurasi sebuah Hugging Face Space baru. Seluruh file yang telah disiapkan kemudian diunggah ke repositori dari Space tersebut. Saat proses build dimulai, platform akan secara otomatis melakukan instalasi dependensi berdasarkan file *requirements.txt*. Setelah semua dependensi terpenuhi, aplikasi web akan diluncurkan dengan menjalankan skrip *app.py*. Dengan demikian, model NER yang telah dibangun menjadi sebuah aplikasi interaktif yang siap digunakan oleh pengguna melalui tautan publik.

3.2 Implementasi

3.2.1 Pengolahan Data CV dan Pseudo-Labeling

A. Ekstraksi Dataset Awal

Pseudocode 3.1 di bawah ini merinci langkah-langkah untuk mengekstrak file-file dari sebuah arsip ZIP ke dalam direktori kerja. Proses ini adalah tahap awal untuk mempersiapkan data mentah yang belum teranotasi. Langkah pertama adalah mendefinisikan lokasi file ZIP dan nama direktori tujuan. Selanjutnya, sistem akan memastikan direktori tujuan tersebut ada, dan jika belum, akan membuatnya secara otomatis. Langkah inti terjadi pada baris ke-4 hingga ke-6, di mana program mencoba membuka file ZIP dan mengekstrak seluruh isinya ke

direktori yang telah disiapkan. Proses ini juga dilengkapi dengan penanganan *error* untuk mengantisipasi jika file tidak ditemukan atau arsip rusak.

Input: Zip file

Output: File yang sudah di unzip

```
1.      function extract_dataset(zip_path, extract_dir):
2.      `// Memastikan direktori tujuan ada`
3.      `create directory` extract_dir `if not exists`
4.      `try:`
5.          `open` zip_path `as` zip_file
6.          zip_file.extractall(extract_dir)
7.          `print` "Ekstraksi berhasil"
8.      `catch` FileNotFoundError:
9.          `print` "Error: File ZIP tidak ditemukan"
10.     `catch` BadZipFileError:
11.         `print` "Error: File bukan arsip ZIP yang valid"
```

Pseudocode 3.2.1: Ekstraksi Dataset Awal

B. Identifikasi Tag Entitas dari Data Anotasi

Pseudocode 3.2 menjelaskan proses untuk mengidentifikasi semua label atau tag entitas yang unik dari file JSON hasil anotasi Label Studio. Fungsi ini membaca file `final_fix.json`, yang berisi data teks beserta anotasi yang telah dibuat. Pada baris ke-4, program mulai mengiterasi setiap data (tugas). Di dalam setiap tugas, program akan menavigasi struktur JSON untuk menemukan daftar result yang berisi setiap entitas yang telah dilabeli (baris 7-9). Setiap label yang ditemukan (misalnya, 'NAME', 'SKILL') akan ditambahkan ke dalam sebuah set untuk memastikan tidak ada duplikasi. Terakhir, pada baris ke-12, set tersebut diubah menjadi sebuah list yang diurutkan dan dikembalikan sebagai output.

Input: File anotasi

Output: Daftar (*list*) berisi nama-nama tag unik

```
1.      function get_unique_tags(json_file):
2.      `load` data `from` json_file
3.      unique_tags ← `new empty set`
4.      `for each` task `in` data:
5.          `if` task `has` 'annotations':
6.              `for each` annotation `in` task['annotations']:
7.                  `if` annotation `has` 'result':
```

```

8.         `for each` result `in` annotation['result']:
9.             labels_list ← result['value']['labels']
10.        unique_tags.add_all(labels_list)
11.        `print` "Error: File bukan arsip ZIP yang valid"
12.        `return sorted list from` unique_tags

```

Pseudocode 3.2: Ekstraksi Tag Unik dari Data Anotasi

C. Pelatihan Model Named Entity Recognition (NER)

Pseudocode 3.3 merinci alur utama untuk melatih model Transformer (RoBERTa) untuk tugas *Named Entity Recognition* (NER). Proses dimulai dengan mendefinisikan semua konfigurasi, termasuk path data, nama model, dan daftar label (baris 1-3). Selanjutnya, program membuat pemetaan label ke ID dengan skema BIO dan memuat tokenizer dari model pre-trained (baris 4-5). Data anotasi dari file JSON kemudian dibaca dan diproses: teks ditokenisasi dan setiap token diberi label BIO yang sesuai (baris 6-7). Dataset lalu dibagi menjadi data latih dan validasi (baris 8). Model RoBERTa untuk klasifikasi token diinisialisasi dengan konfigurasi label yang telah dibuat (baris 9). Setelah argumen pelatihan didefinisikan, Trainer dari library Hugging Face dikonfigurasi dan proses pelatihan dijalankan (baris 10-12). Terakhir, model yang telah dilatih beserta tokenizer dan pemetaan labelnya disimpan ke disk untuk digunakan pada tahap prediksi.

Input: File anotasi

Output: Model NER yang telah dilatih

```

1.    // Konfigurasi
2.    set data_path, model_checkpoint, custom_labels
3.    // Inisialisasi
4.    create label_mappings (BIO scheme: B-TAG, I-TAG, O) from custom_labels
5.    load tokenizer from model_checkpoint
6.    // Proses Data
7.    processed_data ← load_and_preprocess_data(data_path, tokenizer, label_mappings)
8.    datasets ← split_data(processed_data, test_size=0.1)
9.    // Setup Model dan Pelatihan
10.   model ← initialize_model_for_token_classification(model_checkpoint,
label_mappings)
11.   training_args ← define_training_arguments(epochs=10, learning_rate=2e-5)
12.   trainer ← initialize_trainer(model, training_args, datasets)
13.   // Latih dan Simpan Model
14.   trainer.train()

```

```

15.     trainer.save_model(output_dir)
16     save_label_mappings to file

```

Pseudocode 3.3: Pelatihan Model NER

D. Pra-Anotasi (Prediksi) pada Data Baru

Pseudocode 3.4 menggambarkan proses penggunaan model NER yang telah dilatih untuk melakukan pra-anotasi pada CV yang belum memiliki label. Pertama, model, tokenizer, dan pemetaan label yang tersimpan dimuat dari disk (baris 2-3). Sistem kemudian mengiterasi setiap file .txt di dalam direktori *unannotated_cvs* (baris 5). Untuk setiap file, teksnya dibaca dan dimasukkan ke dalam model untuk mendapatkan prediksi entitas (baris 6-7). Hasil prediksi yang berupa token-token berlabel BIO kemudian diolah kembali untuk digabungkan menjadi entitas utuh (misal, [B-NAME, I-NAME] menjadi satu entitas NAME). Terakhir, semua hasil prediksi dari seluruh file diformat ke dalam struktur JSON yang sesuai dengan format impor Label Studio dan disimpan sebagai satu file *ls_bulk_predictions.json* (baris 10-12).

Input: Direktori file yang belum di anotasi & model yang telah dilatih

Output: File prediksi

```

1.     // Muat Artefak
2.     model, tokenizer ← load_fine_tuned_model
3.     id2label_map ← load_label_mapping
4.     // Proses Prediksi Massal
5.     for each cv_file in unannotated_directory:
6.         cv_text ← `read_text_from`(cv_file)
7.         predicted_entities ← `predict_entities`(cv_text, model, tokenizer,
            id2label_map)
8.         `// Format hasil untuk Label Studio`
9.         formatted_prediction ← `format_for_label_studio`(cv_text,
            predicted_entities)
10.        `append` formatted_prediction `to` all_tasks
11.        // Simpan Hasil
12.        save all_tasks as bulk_import.json

```

Pseudocode 3.4: Proses Pra-Anotasi untuk Label Studio

BAB 4 Hasil dan Pembahasan

Bab ini membahas hasil yang diperoleh dari proses pelatihan dan evaluasi model Named Entity Recognition (NER) yang dikembangkan untuk mengekstraksi informasi penting dari dokumen Curriculum Vitae (CV) mahasiswa. Pembahasan meliputi proses pengolahan data, evaluasi model Pseudo-Labeling, evaluasi model SpaCy NER, perbandingan performa antar model, pengaruh variasi hyperparameter, serta kendala dan potensi sistem yang dihasilkan.

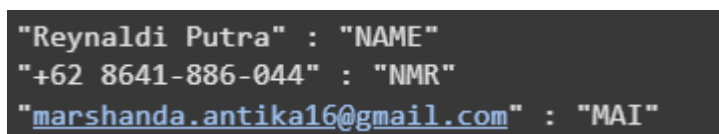
4.1 Hasil Pengolahan Data

4.1.1 Pengumpulan dan Seleksi Data

Langkah awal dalam persiapan data adalah mengumpulkan korpus CV yang beragam dari tiga sumber utama, yaitu Suara Surabaya (50 data), Mabacup ITS (55 data), dan Schematics ITS (109 data), sehingga total terkumpul 214 data mentah. Selanjutnya, dilakukan proses pembersihan manual untuk menyaring data yang paling layak digunakan. Sebuah data dianggap layak jika memenuhi kriteria seperti kelengkapan informasi, format hasil pembacaan PDF yang bersih, dan tidak mengandung nilai yang hilang (NaN). Setelah melalui proses seleksi yang ketat ini, terpilihlah 120 data CV berkualitas tinggi yang akan digunakan pada tahap selanjutnya.

4.1.2 Masking Data

Mengingat CV mengandung Informasi Identitas Pribadi (PII) yang sensitif, tahap masking menjadi sangat penting untuk melindungi privasi kandidat.¹ Proses ini dilakukan secara otomatis menggunakan skrip Python yang mengganti data asli dengan data palsu dari kamus yang telah disiapkan. Data palsu untuk tiga entitas paling sensitif (NAME, PHONE, dan MAIL) digenerasi sebelumnya menggunakan bantuan *Large Language Model* (LLM) Gemini 2.5 Pro untuk memastikan data pengganti terlihat realistis namun tidak merujuk pada individu nyata.



```
"Reynaldi Putra" : "NAME"  
"+62 8641-886-044" : "NMR"  
"marshanda.antika16@gmail.com" : "MAI"
```

Gambar 4.1 Hasil akhir *Masking Data*

Gambar 4.1 memperlihatkan contoh hasil akhir dari proses masking yang telah diterapkan pada data CV.

4.1.3 Anotasi Data Manual

Untuk menciptakan ground truth yang berkualitas, dilakukan proses anotasi data secara manual menggunakan platform Label Studio. Dari total data yang tersedia, sebanyak 80 data (sekitar 67%) dipilih untuk dianotasi. Proses ini dilakukan secara kolaboratif, di mana data dibagi rata kepada empat anggota tim untuk memastikan konsistensi dan efisiensi dalam pelabelan.

4.1.4 Hasil Pseudo-Labeling

Sebanyak 80 data awal telah dianotasi secara manual dan digunakan untuk melatih model awal berbasis BERT dengan bantuan SpaCy. Model ini kemudian digunakan untuk menganotasi data tanpa label secara otomatis melalui teknik Pseudo-Labeling. Setelah diverifikasi manual, data yang valid ditambahkan ke dataset, sehingga jumlah total data berlabel meningkat menjadi 120 data.

4.2 Skenario Pengujian

4.2.1 Desain dan alur pengujian

Pengujian dilakukan dalam tiga kondisi:

- Kondisi Base: Model diuji dengan konfigurasi default.
- Kondisi Hyperparameter Tuning: Model diuji dengan variasi hyperparameter seperti learning rate dan batch size.
- Kondisi Perbandingan Model: Perbandingan antara model BERT BASE dan Gliner pada dataset yang sama.

4.2.2 Dataset dan Konfigurasi Pengujian

Dataset dibagi menjadi data pelatihan dan data validasi dengan perbandingan 1:4.

Model dilatih menggunakan:

- Epoch: 1500
- Learning rate: 1500
- Batch size: 4096
- Tools: SpaCy, Label Studio, Python, Gemini 2.5 Pro

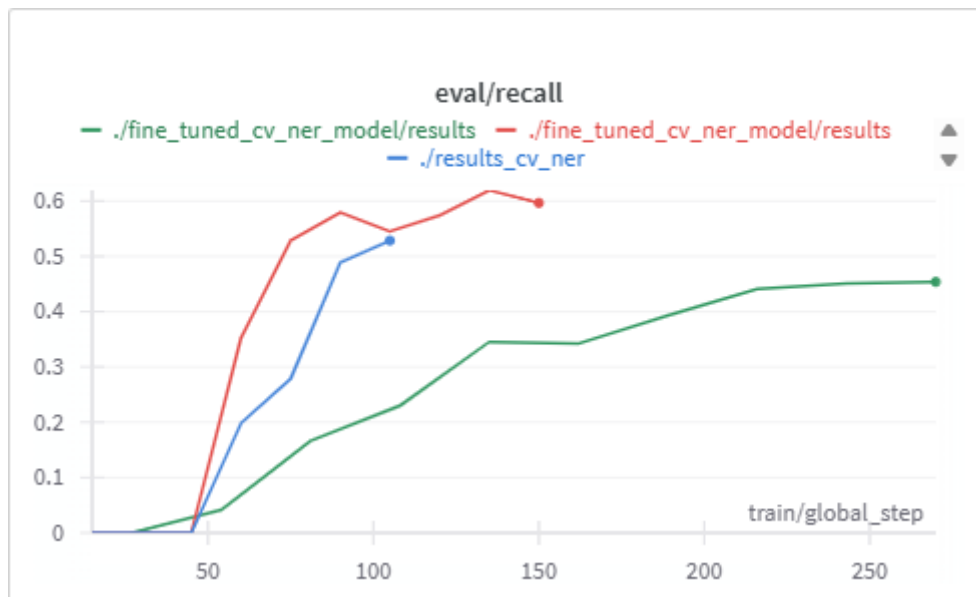
4.3 Hasil Evaluasi Model

4.3.1 Hasil Evaluasi Model Pseudo-Labeling

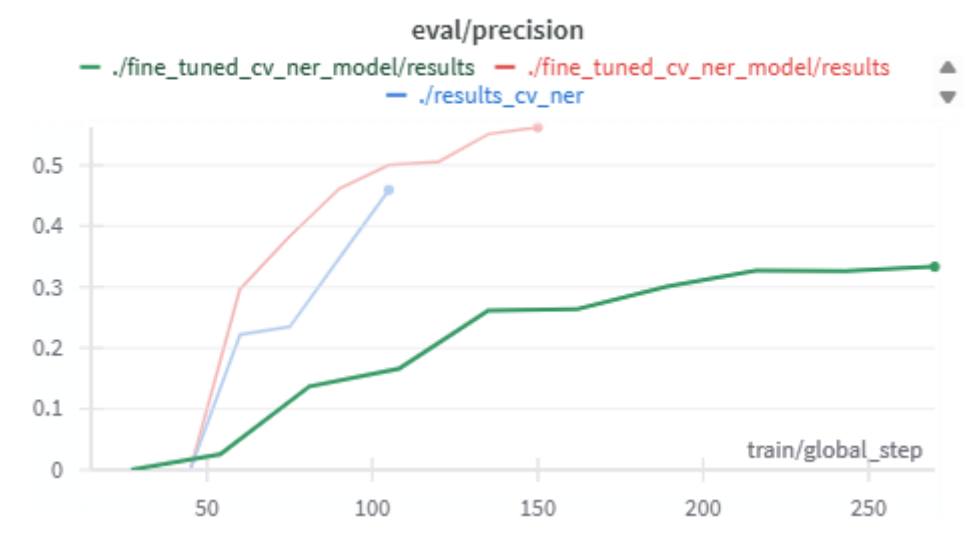
Evaluasi model Pseudo-Labeling dilakukan untuk mengukur efektivitas model dalam membantu proses anotasi otomatis pada dokumen CV. Pengujian dilakukan pada beberapa model dengan perbandingan konfigurasi. Nilai precision, recall, dan F1-score yang dianalisis merupakan hasil rata-rata dari seluruh langkah pelatihan.

Metrik	Nilai Rata - Rata
Precision	0.38
Recall	0.50
F1-Score	0.43

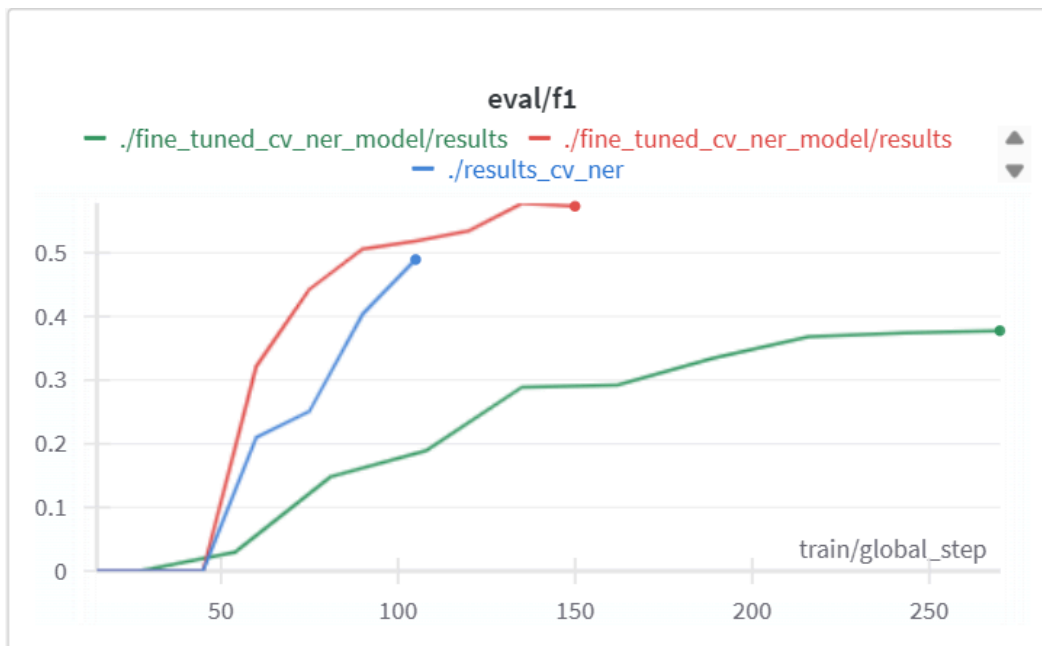
Tabel 4.1 Hasil Evaluasi Model Pseudo-Labeling



Gambar 4.2 peningkatan performa *recall* selama proses pelatihan



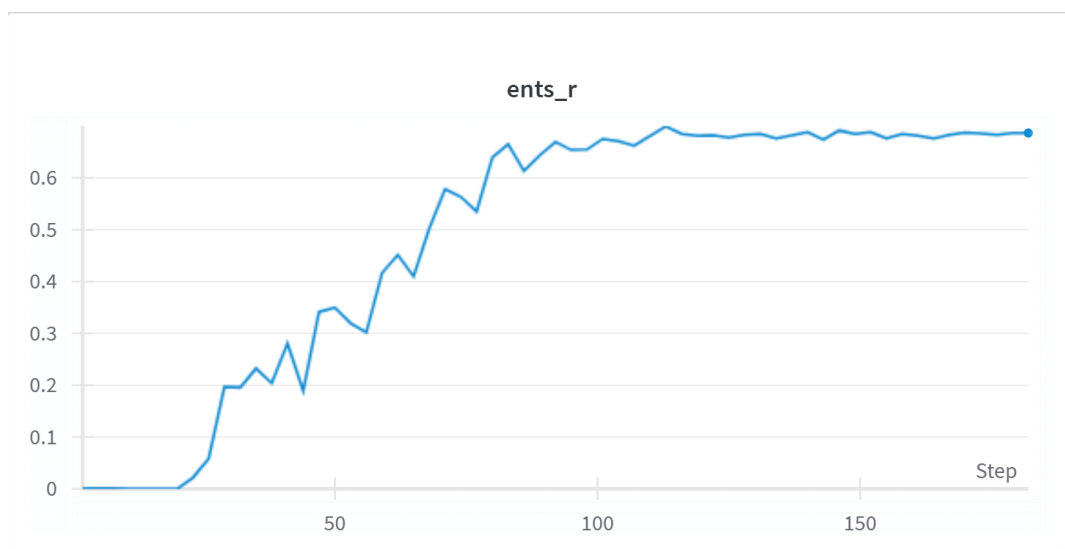
Gambar 4.3 peningkatan performa *precision* selama proses pelatihan



Gambar 4.4 Grafik Evaluasi *F1-Score* Model Pseudo-Labeling

4.3.2 Hasil Evaluasi Model SpaCy NER

Evaluasi performa model Named Entity Recognition (NER) dilakukan untuk mengukur tingkat keberhasilan sistem dalam mengekstraksi informasi dari dokumen CV mahasiswa. Evaluasi dilakukan dengan mengukur rata-rata precision, recall, dan F1-score secara keseluruhan terhadap seluruh entitas yang diekstraksi. Model dilatih selama 1500 epoch untuk memastikan proses pembelajaran yang optimal dan stabil.

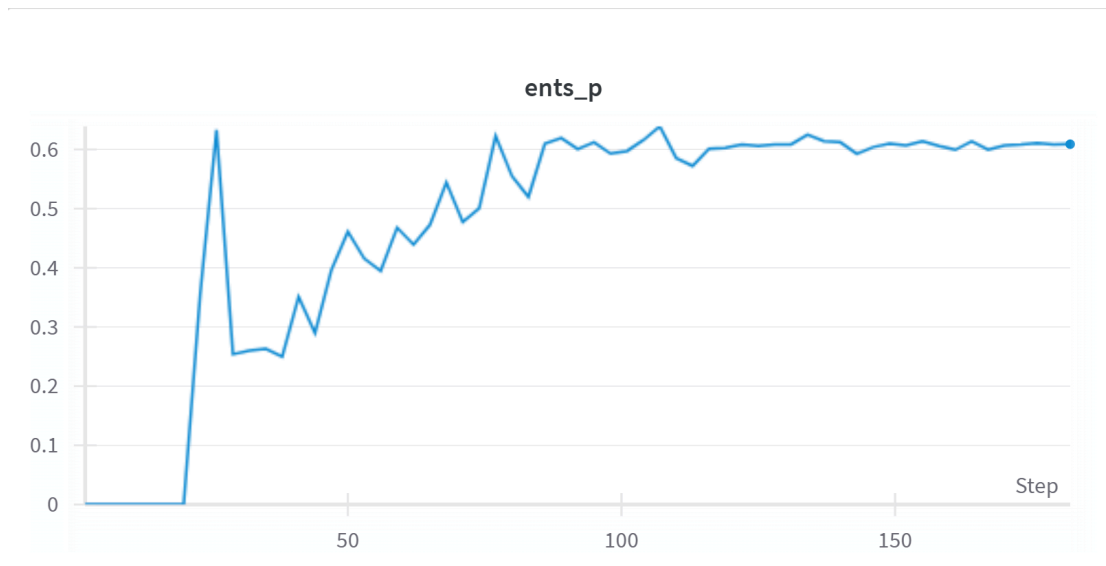


Gambar 4.5 perkembangan nilai *recall* pada model selama proses pelatihan

Gambar 4.5 menunjukkan perkembangan nilai recall (*ents_r*) pada model selama proses pelatihan. Pada tahap awal, recall model masih berada pada tingkat yang rendah dan

menunjukkan fluktuasi. Namun, setelah langkah ke-50, recall mulai menunjukkan peningkatan yang lebih stabil dan konsisten hingga mencapai nilai di atas 0.6.

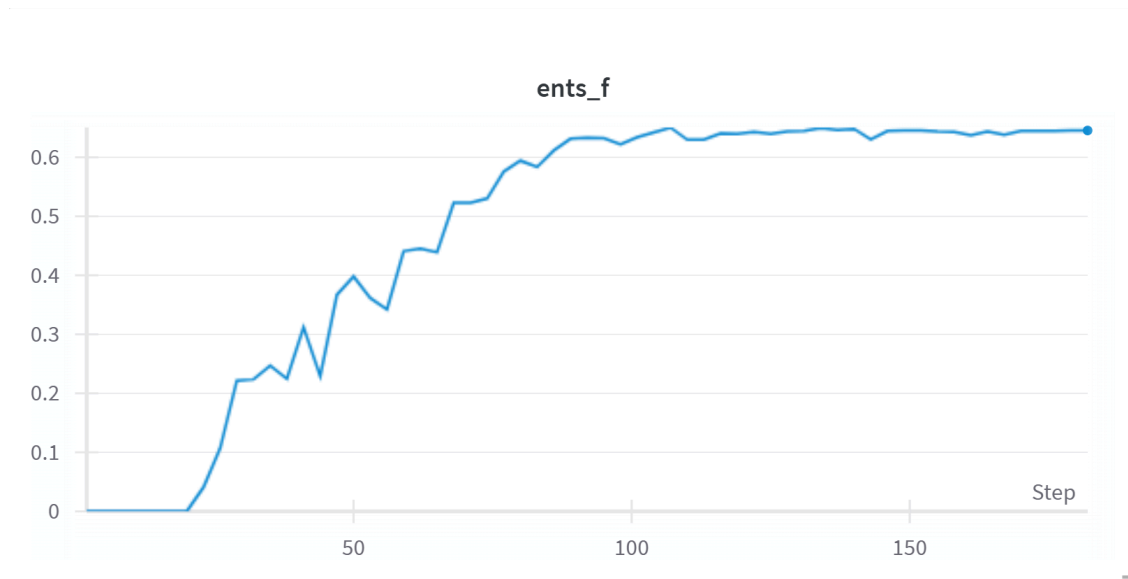
Performa recall yang stabil pada tahap akhir pelatihan mengindikasikan bahwa model semakin mampu mengenali entitas yang relevan secara menyeluruh dalam dokumen CV. Ini berarti model tidak hanya menghasilkan prediksi yang tepat, tetapi juga meminimalisir jumlah entitas yang terlewat.



Gambar 4.6 perkembangan *precision* pada model NER selama proses pelatihan

Gambar 4.6 menunjukkan perkembangan nilai precision (*ents_p*) pada model Named Entity Recognition (NER) selama proses pelatihan. Terlihat bahwa pada tahap awal pelatihan, model mengalami fluktuasi yang cukup signifikan dengan lonjakan tajam di sekitar langkah ke-10. Setelah lonjakan tersebut, precision model sempat mengalami beberapa penurunan namun secara keseluruhan menunjukkan tren peningkatan yang stabil.

Precision model terus meningkat hingga mencapai nilai di atas 0.6 pada langkah-langkah akhir pelatihan. Pola ini menunjukkan bahwa model membutuhkan beberapa siklus pembelajaran untuk mampu mengenali pola entitas secara lebih tepat. Meskipun terdapat ketidakstabilan pada fase awal, model akhirnya mampu beradaptasi dan menghasilkan prediksi dengan tingkat ketepatan yang cukup tinggi.



Gambar 4.7 peningkatan *F1-score* sepanjang proses pelatihan

Gambar 4.7 menampilkan tren peningkatan F1-score (ents_f) sepanjang proses pelatihan. F1-score mengalami peningkatan yang konsisten dan signifikan, terutama pada 100 langkah pertama. Setelah melewati langkah ke-100, performa F1-score mulai stabil dan tetap berada pada kisaran diatas 0.6 hingga akhir pelatihan.

Stabilitas F1-score pada tahap akhir menunjukkan bahwa model tidak hanya mampu membuat prediksi yang tepat (precision), tetapi juga berhasil mengenali sebagian besar entitas yang relevan (recall). Hal ini mengindikasikan bahwa keseimbangan antara precision dan recall telah tercapai dengan baik, yang penting dalam evaluasi ekstraksi informasi.

4.4 Perbandingan Model dan Hyperparameter

4.4.1 Perbandingan Model BERT dan Gliner

Berdasarkan hasil pengujian yang telah dilakukan, diperoleh perbandingan kinerja antara model BERT dan Gliner sebagaimana disajikan pada Tabel berikut:

Model	Precision	Recall	F1-Score
BERT	0.60	0.68	0.64
Gliner	0.45	0.39	0.37

Tabel 4.2 Perbandingan Model BERT dan Gliner

Berdasarkan hasil pengujian model, diperoleh bahwa model BERT secara konsisten menunjukkan performa yang lebih baik dibandingkan model Gliner dalam hal precision, recall, dan F1-score.

Model BERT berhasil mencapai nilai precision sebesar 0.60, yang menunjukkan bahwa sebagian besar entitas yang diprediksi benar oleh BERT memang relevan. Selain itu, nilai recall sebesar 0.68 menandakan bahwa BERT mampu menemukan sebagian besar entitas yang seharusnya dikenali. Gabungan dari precision dan recall ini menghasilkan F1-score sebesar 0.64, yang mencerminkan keseimbangan performa BERT dalam mengenali entitas dengan benar dan lengkap.

Sementara itu, model Gliner menunjukkan performa yang lebih rendah, dengan precision sebesar 0.45, recall sebesar 0.39, dan F1-score sebesar 0.37. Ini menunjukkan bahwa Gliner memiliki kecenderungan lebih tinggi dalam menghasilkan prediksi yang tidak tepat dan gagal mengenali sejumlah besar entitas yang sebenarnya ada.

Perbedaan ini mengindikasikan bahwa model BERT lebih andal dalam menangani tugas Named Entity Recognition (NER), kemungkinan karena arsitektur transformer-nya yang lebih baik dalam memahami konteks dan relasi antar kata dibandingkan pendekatan yang digunakan oleh Gliner.

4.4.2 Pengaruh Variasi Hyperparameter

Eksperimen ini bertujuan untuk menganalisis dampak variasi beberapa hyperparameter utama seperti *batch size*, *learning rate*, *dropout*, dan *warmup steps* terhadap performa model *Named Entity Recognition* (NER). Hasil Percobaan dirangkum pada Tabel 4.3.

Run	Batch Size	Learning Rate	Warmup	Drop out	Precision	Recall	F1-Score	Window	Stride	Max out Pieces	Hidden Width	Acc. Grad.	L2 Reg.	Beta 1	Grad Clip
19	64	1.21e-4	500	0.1668	0.619	0.648	0.633	256	96	3	256	4	2.76e-6	0.8333	4.14
18	64	1.78e-4	250	0.1637	0.625	0.671	0.647	384	96	3	256	4	2.57e-6	0.8559	2.97
17	128	4.33e-4	500	0.1484	0.629	0.676	0.652	384	96	3	256	4	5.32e-6	0.8093	2.50
16	128	1.44e-4	250	0.0556	0.584	0.657	0.618	384	96	2	256	4	3.11e-6	0.8293	4.12
15	64	3.18e-4	500	0.0698	0.638	0.689	0.662	256	96	3	256	4	3.16e-6	0.8139	4.91

Tabel 4.3 Variasi Hyperparameter

Lima konfigurasi diuji untuk mengukur pengaruh variasi hyperparameter terhadap kinerja model NER. Evaluasi dilakukan berdasarkan Precision, Recall, dan F1-Score.

Hasil terbaik diperoleh pada Run 15 dengan F1-Score tertinggi 0.662, menggunakan:

- Batch size 64

- Learning rate 3.18e-4
- Dropout 0.0698
- Window 256
- Maxout pieces 3

Beberapa temuan penting:

- Dropout sedang (0.06–0.16) memberi performa lebih stabil. Dropout terlalu rendah (run 16) menurunkan F1-Score.
- Batch size 64 umumnya menghasilkan skor lebih tinggi dibanding 128.
- Maxout 3 konsisten lebih baik dibanding 2.
- Learning rate tinggi (3e-4 s/d 4e-4) bekerja baik jika diimbangi dropout dan regulasi yang tepat.

Dengan demikian, konfigurasi ringan dan dropout seimbang memberi hasil optimal untuk tugas NER pada data CV.

4.5 Integrasi dengan LLM

4.5.1 Tujuan Integrasi

Integrasi model ke dalam Large Language Model (LLM) bertujuan untuk membantu perekrut (recruiter) dalam mengambil keputusan yang lebih tepat berdasarkan informasi yang terkandung dalam Curriculum Vitae (CV) kandidat. Melalui pendekatan ini, sistem tidak hanya mengekstrak informasi penting, tetapi juga mampu memberikan rekomendasi divisi atau posisi yang sesuai beserta penjelasannya, berdasarkan konteks kemampuan kandidat. Sebagai contoh, jika sistem mendeteksi bahwa kandidat memiliki kemampuan *public speaking*, maka LLM akan menyarankan kandidat untuk ditempatkan di divisi Event, disertai dengan justifikasi yang relevan.

4.5.2 Model Yang Diintegrasikan

Model utama yang digunakan dalam proses ini adalah Named Entity Recognition (NER) berbasis BERT, yang telah dilatih untuk mengekstrak entitas penting dari CV mahasiswa, seperti:

- Nama
- Email
- Nomor Telfon
- Institusi
- Pendidikan
- Jabatan
- Gelar
- Keterampilan (*skills*)
- Bahasa
- Tanggal
- Durasi
- Penghargaan

Output dari model NER ini berbentuk daftar entitas terstruktur yang dapat digunakan sebagai bahan pertimbangan oleh LLM dalam memberikan saran lebih lanjut.

4.5.3 Skema Integrasi Sistem

Proses integrasi dilakukan secara berurutan melalui beberapa tahapan, sebagai berikut:

1. Input CV: Pengguna (recruiter) memberikan CV dalam bentuk teks.
2. Proses NER: Model BERT melakukan ekstraksi informasi untuk mengenali entitas penting dari CV tersebut.
3. Pembuatan Prompt: Hasil entitas yang terdeteksi dijadikan input dalam bentuk prompt ke LLM.
4. Respons LLM: LLM menghasilkan rekomendasi divisi yang sesuai, lengkap dengan alasannya.

BAB 5 Kesimpulan dan Saran

5.1 Kesimpulan

Penelitian ini berhasil mengembangkan sebuah sistem otomatis untuk ekstraksi informasi dari CV mahasiswa menggunakan Named Entity Recognition (NER) berbasis arsitektur Transformer, yang terbukti mampu mengatasi tantangan data tidak terstruktur. Melalui metodologi sistematis yang mencakup pengumpulan 120 CV, anotasi manual, masking data, serta penerapan teknik Pseudo-Labeling yang efektif untuk mengatasi keterbatasan data, model yang dihasilkan menunjukkan kinerja solid dengan F1-Score keseluruhan mencapai 64.54% dan akurasi sangat tinggi untuk entitas krusial seperti Email (0.941) dan Nama (0.847). Sistem ini tidak hanya andal dalam mengidentifikasi data penting, tetapi juga menawarkan potensi implementasi yang fleksibel, baik sebagai aplikasi mandiri maupun sebagai fitur yang terintegrasi dengan Large Language Model (LLM) untuk analisis lebih lanjut, sehingga menyediakan solusi yang efektif bagi perekrut untuk mengubah CV menjadi data terorganisir yang siap dianalisis.

5.2 Saran

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk pengembangan dan penelitian lebih lanjut:

1. Penggunaan Dataset yang Lebih Luas dan Bervariasi

Dataset yang digunakan dalam penelitian ini masih terbatas pada sejumlah dokumen CV. Untuk meningkatkan kemampuan generalisasi model, disarankan menggunakan dataset yang lebih besar, mencakup beragam format dokumen dan variasi gaya penulisan, serta mencakup lebih banyak entitas dan domain industri.

2. Relation Extraction

Penelitian selanjutnya disarankan untuk mengeksplorasi Relation Extraction (RE), yaitu proses mengidentifikasi hubungan antar entitas dalam teks, seperti hubungan kerja atau kepemilikan. Dengan menambahkan RE setelah proses Named Entity Recognition (NER), sistem dapat membangun struktur informasi yang lebih bermakna, seperti knowledge graph. Hal ini akan meningkatkan kualitas ekstraksi informasi dan membuka peluang untuk pengembangan aplikasi lanjutan seperti sistem tanya-jawab dan analisis dokumen otomatis.

3. Peningkatan Evaluasi Berdasarkan Kategori Entitas

Penelitian lanjutan juga dapat memperdalam analisis terhadap performa model pada tiap jenis entitas. Dengan begitu, dapat diketahui entitas mana yang paling sulit dideteksi, dan pengembangan khusus bisa difokuskan pada bagian tersebut.

4. Peningkatan Evaluasi untuk Hasil LLM

Penelitian lanjutan juga dapat memperdalam evaluasi terhadap kualitas saran yang dihasilkan LLM berdasarkan aspek-aspek seperti relevansi, kejelasan alasan, dan kesesuaian dengan data asli. Dengan analisis yang lebih terstruktur, dapat diidentifikasi pola kesalahan atau kelemahan LLM dalam memberikan rekomendasi, sehingga pengembangan sistem dapat difokuskan pada peningkatan akurasi saran dan interpretabilitas respons.

DAFTAR PUSTAKA

- Aggarwal, C. C., & Zhai, C. (Eds.). (2012). *Mining text data*. Springer. <https://doi.org/10.1007/978-1-4614-3223-4>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). *spaCy: Industrial-strength Natural Language Processing in Python* (Version 3.0.0) [Software]. Zenodo. <https://doi.org/10.5281/zenodo.1212303>
- Kumar, A. (2020). A study on pre-processing techniques for optical character recognition. *International Journal of Computer Applications*, 174(41), 26–31. <https://doi.org/10.5120/ijca2020920427>
- Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 260–270. <https://doi.org/10.18653/v1/N16-1030>
- Lee, D.-H. (2013). *Pseudo-Labeling: The simple and efficient semi-supervised learning method for deep neural networks*. Proceedings of the ICML 2013 Workshop on Challenges in Representation Learning (WREPL).
- Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1), 3–26. <https://doi.org/10.1075/li.30.1.03nad>
- Patel, S. (2021). A review on automated information extraction from resumes. *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 1283–1289. <https://doi.org/10.1109/ICAIS50930.2021.9395891>
- Rizvi, S. M., Milios, E. E., & Keselj, V. (2019). Semi-supervised learning for named entity recognition in scientific domains. *Journal of Natural Language Engineering*, 25(2), 217–243. <https://doi.org/10.1017/S135132491800017X>
- Sánchez, D., Batet, M., & Valls, A. (2018). Anonymization of textual data for privacy-preserving data mining. *Data Mining and Knowledge Discovery*, 32(3), 798–826. <https://doi.org/10.1007/s10618-018-0559-0>
- Smith, R. (2007). An overview of the Tesseract OCR engine. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2, 629–633. <https://doi.org/10.1109/ICDAR.2007.4376991>

Affonso, C., Rossi, R. G., & de Paiva, A. C. (2020). Extracting structured data from curricula vitae using natural language processing techniques. *Expert Systems with Applications*, 139, 112851. <https://doi.org/10.1016/j.eswa.2019.112851>

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171–4186). <https://doi.org/10.48550/arXiv.1810.04805>

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 260–270). <https://doi.org/10.48550/arXiv.1603.01360>

Luan, Y., He, L., Ostendorf, M., & Hajishirzi, H. (2018). Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 3219–3232). <https://doi.org/10.48550/arXiv.1808.09602>

Pires, T., Schlinger, E., & Garrette, D. (2019). How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4996–5001). <https://doi.org/10.48550/arXiv.1906.01502>

Purwarianti, A., & Firdaus, A. (2020). Named entity recognition for Indonesian language using BERT model. In *2020 8th International Conference on Information and Communication Technology (ICoICT)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICoICT49345.2020.9154101>

Ravishankara, A., Reddy, S., & Chatterjee, A. (2020). Automated information extraction from resumes using natural language processing. In *2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE)* (pp. 1–5). IEEE. <https://doi.org/10.1109/CISPSSE49931.2020.9212283>

Stanford NLP. (2023). Natural language processing group. <https://nlp.stanford.edu/>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30). <https://doi.org/10.48550/arXiv.1706.03762>

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901. <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html>

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451. <https://doi.org/10.18653/v1/2020.acl-main.747>

GK, G., S, P., & R, S. (2011). Data masking as a tool for data security. *International Journal of Computer Science and Network Security*, 11(6), 143–147. http://paper.ijcsns.org/07_book/201106/20110621.pdf

Goel, V., & Byrne, W. (1999). Minimum Bayes-risk automatic speech recognition. *Computer Speech & Language*, 13(2), 99–115. <https://doi.org/10.1006/csla.1999.0121>

He, P., Liu, X., Gao, J., & Chen, W. (2021). DeBERTa: Decoding-enhanced BERT with disentangled attention. *International Conference on Learning Representations*. <https://openreview.net/forum?id=nZe6z496SP>

Juba, B., & Le, H. S. (2019). Precision-recall versus accuracy and the role of large data sets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1), 4039–4044. <https://doi.org/10.1609/aaai.v33i01.33014039>

Li, X., Li, J., Cui, J., Wang, Z., Yang, Y., & Wang, H. (2019). A deep representation learning model for biomedical named entity recognition. *Journal of Biomedical Informatics*, 99, 103303. <https://doi.org/10.1016/j.jbi.2019.103303>

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, 3, 1–19. <https://doi.org/10.1016/j.aiopen.2022.11.001>

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv. <https://arxiv.org/abs/1907.11692>

Makhoul, J., Kubala, F., Schwartz, R., & Weischedel, R. (1999). Performance measures for information extraction. *Proceedings of the DARPA Broadcast News Workshop*, 249–252. <https://aclanthology.org/R99-1004>

Malik, S., Zha, D., Zhang, Z., & Chen, H. (2024). LLM-Assisted Pseudo-Labeling for Low-Resource Multi-Label Text Classification. arXiv. <https://arxiv.org/abs/2401.02595>

Martin, L., Almahairi, A., Dong, Y., & Botev, V. (2022). Towards efficient NLP: A survey on models, methods, and systems. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 7709–7728. <https://doi.org/10.18653/v1/2022.emnlp-main.524>

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1), 3–26. <https://doi.org/10.1075/li.30.1.03nad>

Sasaki, Y. (2007). *The truth of the F-measure* [Tutorial]. School of Computer Science, University of Manchester. https://www.cs.odu.edu/~kji/spring2017/cs891_IR/papers/F-measure-YS-26Oct07.pdf

- Su, L. T. (1994). The relevance of recall and precision in user evaluation. *Journal of the American Society for Information Science*, 45(3), 207–217. [https://doi.org/10.1002/\(SICI\)1097-4571\(199404\)45:3%3C207::AID-ASI7%3E3.0.CO;2-7](https://doi.org/10.1002/(SICI)1097-4571(199404)45:3%3C207::AID-ASI7%3E3.0.CO;2-7)
- Sun, Y., Zheng, C., Hao, W., & Qiu, X. (2021). *NSP-BERT: A prompt-based zero-shot learner for sentence-level knowledge probing*. arXiv. <https://arxiv.org/abs/2109.00652>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008. <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>
- Yacoub, R., & Axman, D. (2020). A probabilistic F-score for machine learning model evaluation. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9295–9301. <https://doi.org/10.18653/v1/2020.emnlp-main.747>
- Yu, J., & Zhou, G. (2021). A survey on deep learning for imbalanced data classification. *IEEE Access*, 9, 124375–124396. <https://doi.org/10.1109/ACCESS.2021.3110940>