BAB III
NESTED NER DALAM BAHASA INDONESIA

Pada bab ini akan dibahas mengenai pendahuluan mengenai NER, juga teori nested NER juga contoh-contoh penggunaan tugas NER dalam kehidupan sehari-hari. Selain itu, dataset yang digunakan dalam tugas akhir ini juga akan dibahas dengan detail. Penjelasan baik untuk dataset bahasa Inggris yang utama digunakan dari penelitian, maupun dataset bahasa Indonesia yang digunakan dalam penelitian ini. Dataset akan dibahas struktur/bentuk, visualisasi dan juga jenis tagset/jenis entitas yang ditentukan dalam tiap dataset. Bab ini juga ada subbab bagian pra proses (*preprocessing*) dengan rinci untuk mengetahui apa saja yang perlu dimodifikasi dari dataset mentah menjadi dataset yang akhir agar dapat diterima untuk training model.

3.1 Named Enity Recognition (NER)

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity. Istilah named entity awalnya dianggap memiliki relasi dekat dengan pembahasan mengenai *rigid designators* oleh Kripke.[1] Namun diskusi named entity mengarah kepada rigid designators menjadi terlalu filosofis. Sehingga dalam penelitian dicarikan penjelasan yang lebih jelas dan ringkas. Sampai saat ini belum ada persetujuan yang resmi dari bidang NLP mengenai definisi resmi NER. Tetapi dalam skenario penelitian NER dapat disimpulkan definisi umum untuk tugas NER maupun arti sebuah named entities ini. Oleh pihak CoNLL 2002 sendiri, named entities adalah frase yang mengandung nama oleh seseorang, suatu organisasi, sebuah lokasi, sebuah waktu dan sejumlah kuantitas.[2]

[1] Kripke, Saul, Identity and Necessity, M.K. Munitz (ed.). Identity and Individuation. New York: New York University Press, (New York, 1971), pp. 135–64

[2] Kim Sang, Erik F. Tjong, Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition, COLING-02: The 6th Conference on Natural Language Learning 2002, (2002).

Dengan arti yang jelas dan ringkas, tugas pengenalan named entities menjadi persetujuan secara umum definisinya seperti yang telah dinyatakan pihak CoNLL 2002.

Untuk mempermudah dan memperjelas penjelasan, akan dijelaskan dengan perbedaan pengenalan suatu entitas dengan entitas bernama (named entity). Suatu entitas adalah kata-kata yang termasuk kategori jenis entitas namun tidak memiliki suatu nama. Contoh kalimatnya seperti, "Anak kecil itu bermain sepak bola di lapangan". Entitas yang dapat diketahui adalah anak kecil sebagai orang/*person*, lapangan dapat diketahui sebagai lokasi/*location*. Namun dengan tugas NER kedua entitas itu tidak berlaku karena tidak memiliki nama. Untuk tugas NER, kalimat yang tepat sebagai contoh adalah "Budi bermain sepak bola di Lapangan Gelora 10 November". Dengan entitas bernama person untuk Budi, dan entitas bernama jenis location untuk Lapangan Gelora 10 November.

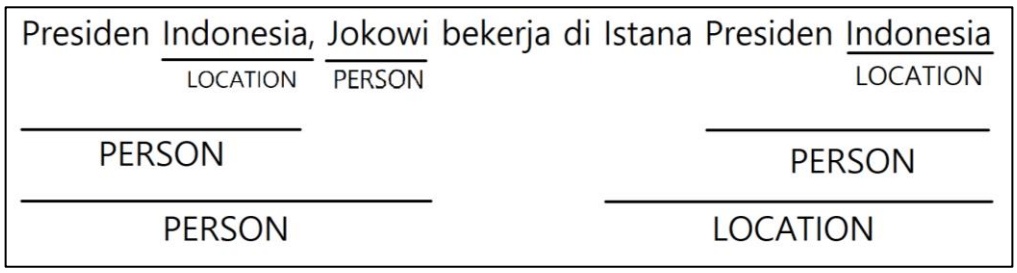
Pada tahun 1997 ada konferensi yang diadakan yang menghasilkan dokumen Message Understanding Conference (MUC-7).[3] Isi dari dokumen ini menjadi salah satu standar umum dalam dunia NLP untuk menentukan kata-kata yang dianggap merupakan named entities dan juga standar jenis-jenis entitas yang akan umumnya sering dipakai ke depannya. Dijelaskan dalam MUC-7 terdapat dianotasikan ekspresi bernama yang unik (seperti person, location, *organization*), ekspresi bilangan (seperti *monetary values*, *percentage*) dan ekspresi waktu (*date, time*). Dan terdapat banyak petunjuk untuk contoh kasus menganotasikan named entities, namun untuk tugas akhir ini tidak seluruhnya mengikuti petunjuk MUC-7 ini. Contohya untuk poin A.1.3, menyatakan jika sebuah entitas yang mengandung substring sebuah entitas bernama yang lain tidak akan dinyatakan entitas bernama sendiri karena tidak dapat dipecahkan. Poin ini tidak berlaku untuk tugas akhir ini karena pada saat ini sedang meneliti metode untuk memecahkan poin tersebut. Contoh dari poin yang dimaksud dalam bentuk kalimat adalah "Arthur Anderson Consulting", dimana Arthur Anderson Consulting merupakan named entity

[3] Chinchor, Nancy, MUC-7 Named Entity Task Definition, (1997).

organization, namun Arthur Anderson tidak dianggap person karena menurut MUC-7 tidak dapat dipisahkan.

Nested NER merupakan task yang tidak jauh berbeda dengan task NER sendiri. Bahkan persentase nested NER pada beberapa dataset signifikan, contohnya pada GENIA terkandung 17% nested NER dan 30% untuk dataset ACE. Namun task ini sering dilewatkan, dimana penelitian pada umumnya hanya fokus pada task NER bukan yang nested/bersarang. Jenny Rose Finkel dan Christopher D. Manning[4] menjelaskan dengan baik mengenai perkiraan alasan mengapa hal ini terjadi. Finkel dan Manning menjelaskan bahwa dipercaya ada dua penyebab NER lebih meningkat daripada nested NER yaitu karena alasan untuk kepraktisan dan alasan teknologi.

Fakta bahwa penelitian terhadap NER disengaja tidak fokus pada nested NER hanya karena kepraktisan adalah karena sebagian besar dataset NER, seperti CoNLL, MUC-6, and MUC-7 NER, memang memutuskan untuk menganotasikan named entity dengan string terpanjang. Semisal untuk kata Bank Indonesia, akan dianotasikan Bank Indonesia dengan named entity organisasi, namun Indonesia tidak akan dianotasikan, di mana seharusnya kata Indonesia seharusnya dianotasikan lokasi. Dan mengapa dipercaya karena teknologi adalah karena tersedianya dataset yang tidak memfasilitasi named entity yang bersarang sehingga banyak penelitian yang mengembangkan untuk pengenalan *flat named entity* (named entity yang tidak bersarang). Contoh nested NER berada pada Gambar 3.1.



Gambar 3.1
Contoh Hasil Task Nested NER

[4] Finkel, Jenny Rose, Manning, Christopher D., Nested Named Entity Recognition, (2009).

Penelitian terhadap nested NER sebelumnya sudah ada seperti penelitian sekitar tahun 2006-2007. Meskipun begitu, nested NER masih belum diperjelas sebagai topik yang perlu diteliti lebih dalam atau topik yang perlu dikembangkan dengan metode selain penggantian dataset.[5] Sehingga Finkel dan Manning melakukan penelitian dengan nested NER sebagai topik dan judul utamanya. Setelah tahun 2009 sedikit demi sedikit penelitian nested NER lebih sering dilakukan meskipun tidak banyak.

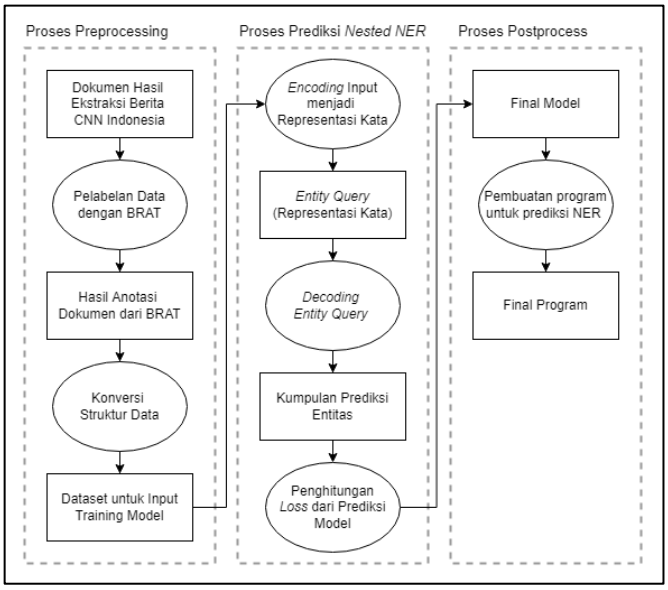
3.2 Arsitektur Sistem

Subbab ini akan menjelaskan arsitektur sistem secara keseluruhan selama pengerjaan tugas akhir. Arsitektur sistem ini akan diulas secara detail pada subbab-subbab berikutnya. Subbab arsitektur sistem akan membahas aliran data dari sub-proses pertama sampai akhir untuk mengetahui gambaran secara umum sistem ini. Visualisasi arsitektur sistem dapat dilihat dari Gambar 3.2 dan dapat dilihat dari gambar tersebut terjadi kurang lebih lima buah proses. Dan untuk setiap beberapa proses telah dikelompokkan menjadi kelompok sub-proses agar dapat dimengerti proses tersebut dilakukan dengan tujuan tertentu. Semisal untuk pra proses, mungkin untuk prediksi NER atau pasca proses untuk programnya.

Gambar 3.2 memiliki pembagian tiga sub-proses, pra proses, proses prediksi nested NER, dan pasca proses. Pra proses adalah proses paling pertama, dan memiliki dua proses yang berhubungan dengan dataset. Proses ini fokus pada memroseskan dataset menjadi input yang sesuai untuk model program. Dari dataset mentah dari ekstraksi berita CNN Indonesia, akan diubah menjadi struktur data sesuai dengan struktur untuk model yang digunakan di tugas akhir ini. Kemudian lanjut kepada sub-proses berikutnya adalah proses prediksi nested NER. Proses yang dilakukan pada sub-proses ini adalah bagian dari model. Metode-metode yang ditentukan untuk training model sehingga dapat melakukan prediksi nested NER. Dan metode ini akan dijelaskan lebih detail pada bab-bab berikutnya. Setelah proses paling akhir proses ini, yaitu penghitungan *loss*, proses ini menghasilkan model

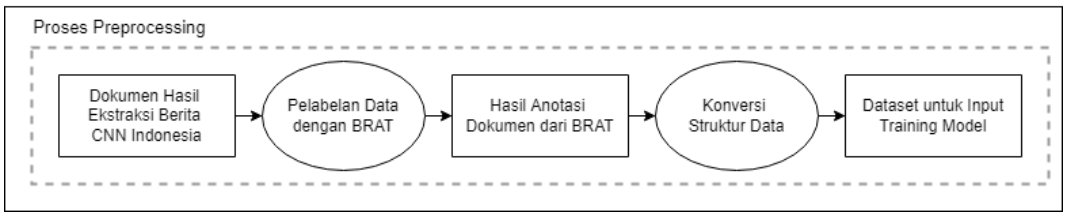
[5] Byrne, Kate, Nested Named Entity Recognition in Historical Archive Text, ICSC '07: Proceedings of the International Conference on Semantic Computing (2007), hal. 589– 596.

final yang telah memiliki informasi dan kemampuan untuk menentukan nested NER. Dan model akhir ini akan menjadi input utama dari proses berikutnya yaitu pasca proses. Sub-proses bagian pasca proses ini bertujuan untuk menyiapkan model yang diterima, untuk menerima input dan output yang dapat dilihat oleh user / orang. Untuk detail dari masing-masing arsitektur sistem akan dijabarkan dalam subbab-subbab berikutnya.



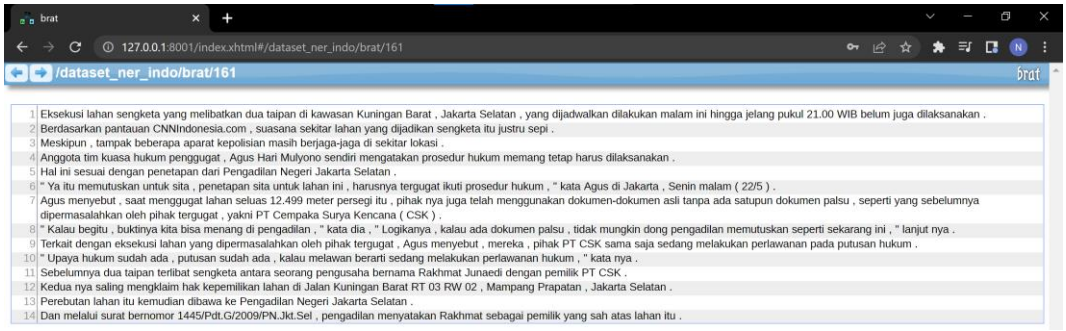
Gambar 3.2
Arsitektur Sistem

Penjelasan berikut akan membahas tahap pra proses yang akan dilakukan dalam arsitektur sistem ini. Pembahasan pada bagian ini akan menjabarkan mengenai proses yang akan dilakukan dalam tahap ini serta input dan output yang akan digunakan dan dihasilkan dalam tahap pra proses ini. Gambaran setiap tahap pra proses dapat dilihat pada Gambar 3.3. Dapat dilihat terdapat dua proses (digambarkan dengan lingkaran) dan tiga data (digambarkan dengan persegi panjang) yang akan digunakan dalam proses ini. Setiap data akan diberikan penggambaran dan akan diberikan penjelasan bagaimana data itu akan diterima atau bagaimana akan diolah.



Gambar 3.3
Arsitektur Sistem Pra Proses

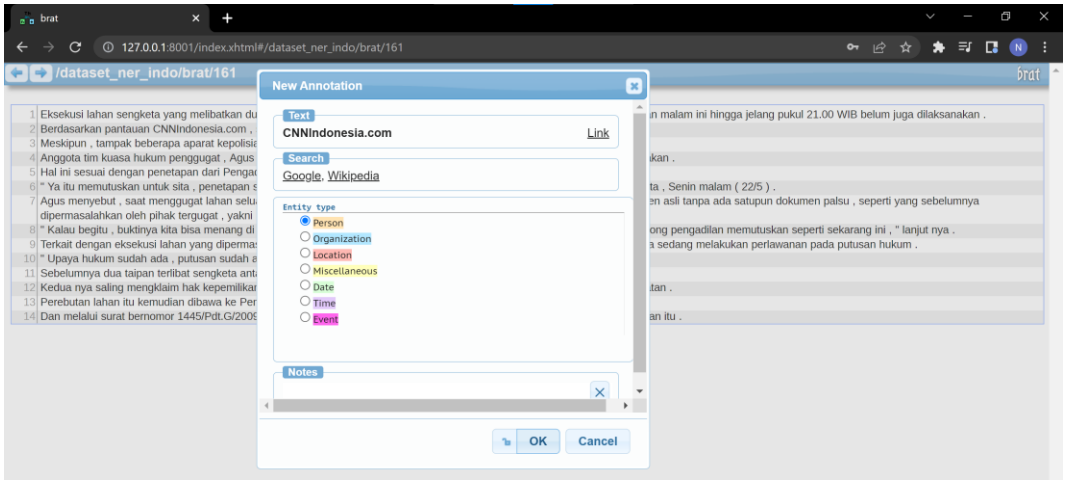
Input pada tahap pertama pra proses adalah hasil ekstraksi berita dari CNN Indonesia. Ekstraksi ini berupa format *text file* (format txt) kemudian data tersebut akan menjadi input untuk proses pelabelan data dengan alat anotasi BRAT. Seperti yang telah dijelaskan, pada bab kedua subbab BRAT, data ini akan diterima dan setiap kalimat akan dipisah dengan fitur *linebreak* dari BRAT sendiri (pemisahan setiap kalimat menjadi 1 baris sendiri dalam sebuah text file). Dan hasil file yang telah diubah ini akan menjadi teks yang muncul pada halaman anotasi BRAT. Isi dari file teks akan dilampirkan dalam halaman BRAT seperti pada Gambar 3.4.



Gambar 3.4
Tampilan File Teks pada BRAT

BRAT dipilih sebagai alat anotasi karena kemudahan kegunaannya dalam pelabelan. Karena penggunaan fitur yang intuitif dan mirip dengan perintah mouse dengan teks pada umumnya, cara pelabelan sebuah / beberapa kata dapat dilakukan dengan menekan dan menggeser *mouse*, memilih kata yang ingin dilabelkan. Kemudian akan muncul *window* jenis-jenis label yang ingin dipilih. Window tersebut juga memiliki fitur lain yang membantu seperti fitur Link untuk menyimpan URL address dari kata yang dianotasikan tersebut untuk menuju ke kata tersebut dengan mudah dan cepat. Fitur search dengan tombol Google atau

Wikipedia untuk membantu mencari arti yang relevan dari kata tersebut. Fitur Notes untuk memberi catatan pada anotasi tersebut. Jika selesai melakukan pelabelan atau fitur lain, cukup tekan tombol OK untuk menyimpan hasil perubahan. Tampilan window ini dapat dilihat pada Gambar 3.5.



Gambar 3.5
Tampilan Window Fitur Pelabelan

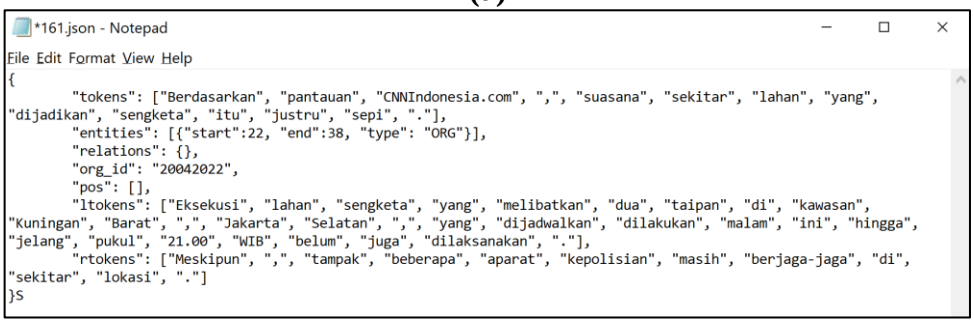
Contoh pada Gambar 3.4 adalah satu file teks/dokumen dari dataset penelitian. Setiap dokumen memerlukan file teks agar dapat dilabelkan, dan untuk pencatatan pelabelan/anotasi yang telah dilakukan akan disimpan dalam jenis file yang dibuatkan sendiri oleh BRAT secara otomatis yaitu jenis file ann (annotation). File teks ini akan menjadi output dari proses pertama yaitu pelabelan data berita CNN Indonesia juga menjadi input untuk proses berikutnya yaitu konversi struktur data. Visualisasi struktur data dari BRAT dan struktur data untuk input training model akan ditampilkan pada Gambar 3.6.

Struktur data dari file teks anotasi BRAT memiliki 4 jenis catatan. Gambar dibawah akan diambil sebagai contoh untuk menjelaskan keempat jenis catatan dari BRAT. Catatan terdiri dari T1, ORG, 212, 228 dan CNNIndonesia.com. T1 merupakan kode pelabelan untuk dokumen tersebut, tiap catatan pelabelan akan mendapatkan kode tersebut dan sifatnya unik dan *incremental* (T1, T2, T3, dst). Catatan ORG adalah jenis label yang dipilih, dalam tugas akhir ini, ORG adalah label/jenis entitas organisasi. Angka 212 dan 228 adalah indeks pertama dan

terakhir dari dokumen itu untuk mengambil kata/huruf yang dilabel untuk pelabelan tersebut. Sehingga jika dari dokumen diambil kata-kata dari 212 sampai dengan 228 akan mendapatkan kata-kata yang sama dengan catatan berikutnya yaitu CNNIndonesia.com.

Dapat dilihat pada Gambar 3.6 (b), jenis filenya struktur data input model bentuk JSON (Java Script Object Notation). Penjelasan atribut dari JSON hanya atribut yang digunakan dalam tugas akhir ini. Tokens adalah *array* kata-kata dalam 1 kalimat yang sedang dianotasikan saat ini. Entities adalah semua label entitas yang ditemukan dalam 1 kalimat itu, dengan bentuk *array of JSONs* yang memiliki 3 atribut. Atribut start dan end adalah index awal dan akhir dari kalimat saat ini, dan type sebagai catatan label/jenis entitas apa untuk kata tersebut. Meskipun tugas akhir ini memiliki batasan tidak menggunakan POS Tag, namun model tetapi meminta POS Tag untuk mengolah data. Karena itu atribut pos itu akan berisi token *unknown* hanya sebagai data pengisi agar tidak memunculkan *error* dalam training program. Ltokens dan rtokens memiliki tujuan yang sama dengan tokens perbedaannya adalah ltoken merupakan kalimat sebelum kalimat saat ini. Dan rtokens adalah kalimat setelah kalimat saat ini.

(a)



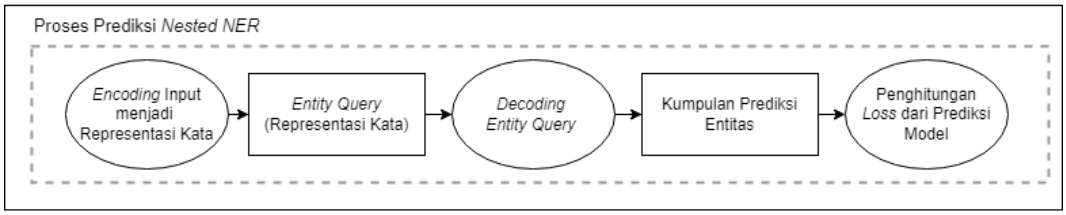
(b)

Gambar 3.6
Struktur Data (a) Data Anotasi BRAT (b) Data Input Model

Dalam tugas akhir ini telah dibuatkan program untuk merubah struktur data BRAT menjadi struktur data Sequence-to-Set, karena tidak adanya program konversi tersebut. Setelah struktur data dari file teks BRAT diubah menjadi struktur data yang dibutuhkan, menghasilkan dataset dengan struktur sesuai untuk menjadi input training model. Dengan ini, penjelasan persiapan dataset dalam bagian pra proses selesai dan dataset tersebut akan dilanjutkan kepada proses berikutnya yaitu proses prediksi nested NER.

Berikut menjelaskan proses prediksi nested NER dengan melakukan training dengan metode Sequence-to-Set Network. Gambar 3.7 menggambarkan visualisasi aliran system untuk prediksi nested NER ini. Terdiri dari tiga proses dan dua data yang digunakan. Untuk detail seperti metode apa yang digunakan dalam tiap proses, modifikasi atau parameter yang digunakan dalam tiap proses tidak akan dijelaskan dalam bab ini namun pada bab-bab kedepannya.

Aliran sistem ini akan dimulai dengan sebuah proses yang menerima input dari proses sebelumnya yaitu tahap pra proses. Input merupakan dataset yang sudah diubah sesuai struktur data yang ditentukan oleh model Sequence-to-Set Network. Proses ini mengambil inputnya dan melakukan encoding, hal ini bertujuan untuk membuat representasi kata yang baru dengan informasi yang terdapat dari input dataset saat itu. Hal ini dicapai dengan bantuan berbagian jenis embedding yang saling digabungkan. Representasi kata yang baru ini dibutuhkan agar membantu computer untuk training model dengan representasi dalam bentuk yang dapat mudah diolah yaitu angka bukan kata-kata.



Gambar 3.7
Arsitektur Sistem Proses Prediksi Nested NER

Representasi kata tersebut merupakan output dan juga input pertama dari sub-proses ini, dimana representasi kata itu akan bersama dengan sebuah set vektor

yang *trainable* yaitu *Entity Query/Queries*. Secara teori, isi dari representasi kata ini adalah gabungan berbagai jenis embedding yang ada. Jenis embedding ini akan dijelaskan pada sub-bab berikutnya. Dan data ini, dari urutan input, memiliki panjang *l x d*, dimana *l* adalah panjang urutan input saat ini, dan *d* sebagai dua kali panjang *hidden size* dari LSTM yang digunakan pada proses encoding (proses sebelumnya).

Lanjut pada proses berikut yang menerima entity queries dan representasi kata (tokens) adalah bagian kedua dari arsitektur sistem dari model ini yaitu proses *decoding* entity queries. Target dari proses ini adalah untuk melakukan membaca informasi yang telah "diringkas" dalam entity queries. Sebagian besar dari pekerjaan *decoder* dilakukan dengan mekanisme Attention yang diambil dari metode Transformer (metode ini telah dijelaskan pada bab sebelumnya mengenai bagian decoder). Decoder akan mempelajari kata-kata yang perlu diperhatikan dan ketergantungan antar entitas untuk mengetahui pola prediksi nested NER. Setelah entity queries dan tokens dilewatkan self-attention dan cross-attention, hasil dari cross-attention akan melewati bagian Feed Forward Network (FFN), bagian ini bertugas untuk lebih mengetahui hubungan antar nilai dalam embedding. Dan output dari FFN akan menjadi input pada layer terakhir bagian decoder yaitu Multilayer Perceptron (MLP) untuk mengklasifikasikan embedding FFN menjadi hasil akhir batasan kiri dan kanan prediksi jenis entitas dan juga jenis entitas yang diprediksikan.

Output dari MLP ini adalah data "kumpulan prediksi entitas" dalam Gambar 3.7. Dan dengan output ini akan memasuki proses paling akhir yaitu penghitungan *loss* untuk menentukan performa dari model yang sedang training. Hal ini dilakukan dengan metode *bipartite matching*. Suatu metode yang sering digunakan untuk membantu penghitungan *assignment matching*. Dalam bagian ini, akan diusahakan untuk mencari nilai loss, dibutuhkan nilai optimal pasangan prediksi entitas dengan target entitas sebenarnya. Nilai optimal tersebut diambil dari algoritma Hungarian. Nilai loss ini akan membantu model untuk mempelajari cara prediksi nested NER.

Hasil dari proses prediksi ini adalah model yang sudah melewati proses training dan memiliki ilmu untuk prediksi nested NER dengan optimal. Setelah

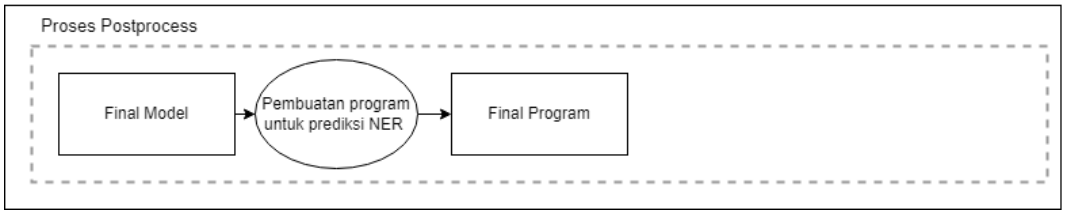
mendapatkan model yang optimal, model ini diperlukan untuk memprediksikan input kalimat yang ingin dicarikan entitas nya. Karena itu perlu dilewatkan tahap pasca proses yang menyiapkan model untuk menjadi program yang dapat digunakan user dengan mudah.

Proses pasca dari bagian prediksi akan dibahas dalam bagian subbab ini. Alur dari proses tersebut dapat dilihat pada Gambar 3.8 bahwa hanya terdapat satu proses yang dijalankan dengan dua hasil proses yang disediakan. Final model merupakan hasil akhir model dengan metode sequence-to-set network yang telah ditraining sedemikian rupa dengan akurasi terbaik. Model ini akan berbentuk file ekstensi .bin dan beberapa file lainnya yang akan menjadi file pendukung untuk menjalankan model tersebut. File tersebut adalah config.json, extra.state, pytorch_model.bin, special_tokens_map.json, tokenizer_config.json, vocab.txt.

Config.json adalah konfigurasi untuk model tersebut saat akan dijalankan untuk prediksi dari input program. Extra.state adalah metadata untuk model mengenai versi model dan data lainnya. Pytorch_model.bin adalah model tersebut yang akan dijalankan dengan bantuan informasi lain seperti konfigurasinya (config.json), token spesial yang akan digunakan dalam proses analisa dataset (special_tokens_map.json), juga kosakata dari kata-kata yang pernah muncul didataset (vocab.txt). File tokenizer_config.json adalah file berisi informasi mengenai tokenizer model yang digunakan untuk perubahan kata-kata dari dataset menjadi token sesuai dengan konfigurasi yang ditentukan. Contoh konfigurasi adalah apakah semua kata-kata dikonversikan menjadi *lower case*, nama dari model tokenizer yang digunakan, token-token spesial yang dapat digunakan tokenizer untuk kata/karakter spesial (kata-kata yang tidak diketahui diberikan *unknown token* biasanya dalam bentuk [UNK]/<UNK>. Seluruh file ini akan digunakan dalam proses berikutnya yaitu prembuatan program prediksi NER dengan final model.

Proses pembuatan program untuk prediksi NER adalah proses dimana file-file dari model yang telah melewati proses training, akan ikut serta dalam pembuatan program baru. Program ini akan menerima input user, kemudian program akan memberikan input ini kepada final model untuk memprediksi tag

named entity dari kalimat input user tersebut. Program ini jika selesai dapat digunakan untuk menerima input beberapa kalimat user (3-4 kalimat), kemudian akan dioutput kan dalam file bentuk .ann, seperti yang pernah disebut, file pelabelan untuk BRAT untuk mempermudah visualisasi.



Gambar 3.8
Arsitektur Sistem Pasca Proses

Program ini akan terdiri dari beberapa file .py (file bahasa pemrogramman Python), dimana file-file tersebut juga memiliki alur sendiri. Pertama program menerima input kalimat dalam bentuk txt, kemudian kalimat tersebut akan dikonversikan menjadi struktur data yang sesuai untuk diterima model agar dapat diprediksi. Setelah dikonversikan, program akan melempar kalimat yag sudah dikonversi kedalam model yang otomatis akan memberikan sejumlah entitas prediksi yang diprediksikan. Program akan mengubah hasil prediksi ini menjadi file .ann agar dapat melihat visualisasi hasil prediksi tersebut dalam program BRAT.

3.3 Dataset dan Tagset Bahasa Inggris

Dataset yang digunakan pada penelitian yang dirujuk di tugas akhir ini menggunakan beberapa dataset, tetapi sebagai rujukan utama di penelitian utama ini akan fokus pada dataset GENIA[6]. Sebagai informasi tambahan, dataset lain yang digunakan oleh peneliti Sequence-to-Set Network adalah ACE 2004[7], ACE 2005[8],

[6] Kim, J.D., dkk., GENIA corpus—a semantically annotated corpus for bio-textmining, Bioinformatics (2003), Vol. 19 Suppl. 1 2003, hal. i180–i182
[7] Shachi Language Research Search, ACE 2004 Multilingual Training Corpus, http://shachi.org/resources/593, 2017.
[8] Linguistic Data Consortium, ACE 2005 Multilingual Training Corpus, http://catalog.Idc.upenn.edu/ldc2006t06, 2018.

KBP 2017[9]. Keempat dataset yang digunakan merupakan beberapa dataset standar yang digunakan dalam penelitian NER bahasa Inggris. Berikut penjelasan singkat untuk dataset yang digunakan dalam Sequence-to-Set Network.

Dataset GENIA adalah dataset utama untuk penulisan biomedis dalam penelitian bidang NLP. Tujuan utama adanya dataset ini untuk membantu perkembangan ekstraksi informasi dan *mining* teks dalam domain *molecular biology*. GENIA dapat digunakan dalam beberapa kategori penelitian seperti *Part-of-Speech annotation, Constituency* (*phrase structure*) *syntactic annotation, Term annotation, Event annotation, Relation annotation, Coreference annotation*. Bahkan GENIA adalah salah satu dataset yang pada awalnya menganotasikan dataset nya secara nested bukan flat (tidak menghiraukan nested named entities).

Sedangkan dataset ACE 2004 dan ACE 2005 berasal dari program riset yang sama yaitu ACE (Automatic Content Extraction) oleh institut NIST (National Institute of Standards and Technology) yang mengutamakan pengembangan ekstraksi informasi dan sudah berjalan dari tahun 1999 sampai 2008. Tidak hanya bahasa Inggris, dataset ACE berkembang tiap tahunnya dengan menyediakan untuk bahasa Mandarin, bahasa Arab, dan bahasa Spanyol. Teks yang disediakan ACE berasal dari teks berita maupun teks siaran.

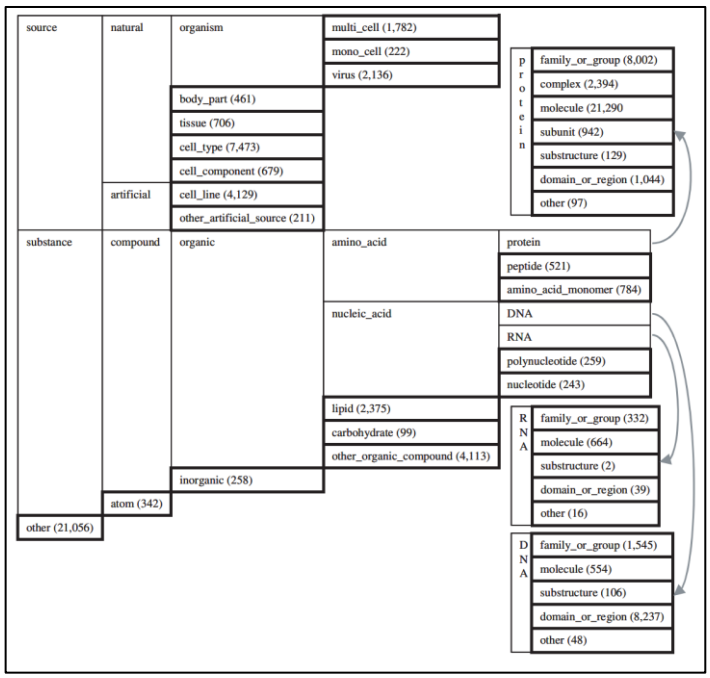
Selain ACE, NIST juga menyediakan dataset lain dari konferensi mereka Text Analysis Conference 2017 dengan nama dataset Knowledge Base Population (KBP) 2017. TAC adalah konferensi yang menyediakan *evaluation workshop* untuk membantu berkembangnya penelitian di bidang NLP dengan menyediakan data dan cara evaluasi kemudian mengadakan forum untuk membahas hasil-hasil yang telah ditemukan oleh peneliti/organisasi lainnya. Ada beberapa jenis task yang disediakan oleh TAC untuk diteliti dan salah satunya adalah Entity Descovery and Linking (EDL) dalam area KBP tersebut. Task selain EDL adalah *Cold Start KB* (CSKB), *Slot Filling* (SF), *Event, Belief and Sentiment* (BeSt) dan area tersendiri *Adverse Drug Reaction Extraction* (ADR). Dataset yang disediakan untuk EDL digunakan sebagai salah satu dataset untuk Sequence-to-Set Network.

[9] National Institute of Standards and Technology, Text Analysis Conference (TAC) 2017, https://tac.nist.gov/2017/index.html, 2017.

3.3.1 GENIA

GENIA, dataset utama biomedis untuk penelitian NLP, telah berkembang dari tahun 1998 sampai saat ini, dengan fokus utama yaitu perkembangan ekstraksi informasi dan mining teks dalam domain molecular biology. Pada 2003, GENIA telah menghasilkan dataset versi ketiga yang terdiri dari 2.000 abstrak, lebih dari 400.000 jumlah kata, dan kurang lebih 100.000 anotasi dengan campur tangan manusia.

Isi dari dataset ini terdiri dari ekstraksi artikel dari MEDLINE. Bentuk dari GENIA adalah artikel yang menggunakan bentuk mark-up berbasis XML dengan ID MEDLINE, judul dan abstrak untuk setiap artikel MEDLINE. Setiap abstrak terdapat teks yang terbagi menjadi kalimat-kalimat. GENIA mengutamakan kualitas dari anotasinya, seluruh abstrak dan judulnya ditentukan oleh dua pakar domain untuk istilah biologis, dan istilah biologis tersebut telah dianotasi secara semantik dengan deskripsi dari ontologi GENIA (dapat dilihat pada Gambar 3.9).



Gambar 3.9
Ontologi dan Statistika dari GENIA[10]

[10] Kim, J.D., dkk., GENIA corpus—a semantically annotated corpus for bio-textmining, Bioinformatics (2003), Vol. 19 Suppl. 1 2003, hal. i180–i182

Tabel 3.1 adalah spesifikasi dataset untuk penelitian nested NER dari GENIA. Penelitian Sequence-to-Set Network menggunakan spesifikasi Tabel 3.1 untuk pembagian data training dan test. Hasil penghitungan ini terdapat dari sejumlah 2000 abstrak yang dimiliki. Dan penghitungan jumlah named entity dan nested named entity dari 18.546 kalimat. Tabel ini menjadi referensi untuk dataset yang digunakan dalam tugas akhir ini dengan pernyataan minimal 10-20% nested entities didapat dari jumlah entities yang dimiliki dataset tersebut.

Tabel 3.1
Spesifikasi Dataset GENIA

Statistika	Jumlah Train	Jumlah Test
Kalimat	16.692	1.854
Kalimat dengan Nested Entities	35.22	446
Rata-rata panjang kalimat	25.35	25.99
Seluruh Entity	50.509	5.506
Seluruh Nested Entities	9.064	1199
Persentase Entities (%)	17.95	21.78

3.3.2 Penjelasan Jenis Tagset Bahasa Inggris

Dataset utama yang digunakan untuk uji coba Sequence-to-Set Network dalam tugas akhir ini adalah GENIA. Namun, tagset yang dirujukkan lebih mengarah kepada tagset yang digunakan ACE dengan beberapa tambahan tagset lain (terinspirasi dari jenis tagset dari *open library* spaCy). Tabel di bawah adalah seluruh tagset yang digunakan untuk anotasi dataset ACE 2005 juga ACE 2004. Pada Tabel 3.2 terdapat kolom entitas yang berisi istilah yang digunakan untuk named entity nya. Istilah dari entitas ada dua jenis yaitu istilah panjangnya dengan istilah pendek, istilah pendek digunakan untuk memudahkan visualisasi dan anotasi saat proses pelabelan. Kemudian kolom keterangan untuk memberi penjelasan singkat mengenai entitas apa yang akan di anotasikan sebagai named entity. Kolom terakhir adalah contoh konkret subjek yang dapat dianotasikan sebagai named entity.

Tabel 3.2
Daftar Tagset ACE 2005

Entitas	Keterangan	Contoh
Person (PER)	Seseorang/kumpulan orang	President Obama
Organization (ORG)	Perusahaan, agensi, grup dalam struktur organisasi	Secret Service
Facility (FAC)	Bangunan, gedung, perumahan	White House
Location (LOC)	Lokasi geografis (darat, laut, dsb)	Red Sea
Geo-Political Entity (GPE)	Wilayah geografis dalam kekuasaan politik atau grup sosial	United States of America
Vehicle (VEC)	Alat untuk memindahkan	Limousine
Weapon (WEA)	Alat untuk menyakiti/menghancurkan	Sniper, Knife

ACE 2004 dan ACE 2004 berada di bawah lisensi dan dataset ini tidak dapat diperoleh secara gratis. Harga yang dicantumkan untuk yang bukan anggota dari organisasinya akan mendapatkan tarif $ 4.000,00. Tabel merepresentasikan ketujuh tagset dari ACE, untuk contoh yang diambil secara langsung dari dataset ACE tidak dapat dilampirkan karena lisensi dari dataset tersebut. Karena itu, untuk beberapa subbab kedepan mengenai penjelasan tagset bahasa Inggris akan menggunakan contoh dan visualisasi dari penulis sendiri.

3.4 Dataset dan Tagset Bahasa Indonesia

Pada saat ini, belum ada dataset umum untuk penelitian nested NER dalam bahasa Indonesia yang dapat digunakan sebagai referensi maupun yang dapat digunakan untuk penelitian sebagai dataset. Karena ini, dataset yang digunakan dalam tugas akhir ini diambil dari tugas akhir sebelumnya yang juga meneliti dalam topik NER, yaitu dataset dari tugas akhir mahasiswa Georgia Nikita (218116685)[11] yang juga menggunakan topik nested NER dalam tugas akhirnya. Dataset ini juga berasal dari tugas akhir sebelum Georgia Nikita yaitu mahasiswa Christian Nathaniel Purwanto (214116299)[12]. Seperti yang diulas dalam tugas akhir Christian Nathaniel Purwanto, bahwa pemilihan tagset/jenis entitas dalam dataset ini

[11] Georgia Nikita, Skripsi: "Service Oriented Nested NER untuk Ekstraksi Keyword Entitas di Portal Berita Bahasa Indonesia" (Surabaya: 2022).
[12] Christian Nathaniel Puerwono, Skripsi: Ekstraksi Entity dan Relasi Dalam Bahasa Indonesia Menggunakan Bidirectional LSTM" (Surabaya: 2018).

mengambil inspirasi dari jenis tagset bahasa Inggris yang ada, namun juga tagset yang dipilih disesuaikan dengan dataset bahasa Indonesia tugas akhir ini.

Dokumen dataset nested NER tugas akhir ini bersumber dari beberapat situs berbeda. Secara singkat, sumber dataset berasal dari berita CNN Indonesia, situs ensiklopedia Wikipedia Indonesia dan crawling beberapa situs dari Google yang telah ditranslasikan ke dalam bahasa Indonesia. Secara keseluruhan, dokumen dataset ini memiliki domain berita politik sehingga entitas yang digunakan telah disesuaikan dengan seringnya beberapa jenis entitas sering disebutkan. Beberapa contoh jenis entitas yang paling sering disebut adalah person, organization, dan date atau time.

3.4.1 Jenis Tagset Bahasa Indonesia

Berikut adalah penjelasan mengenai jenis-jenis entitas/tagset yang ditentukan dalam dataset tugas akhir ini. Terdapat tujuh jenis entitas dan tiap penjelasan dapat ditemukan dari Tabel 3.3. Tabel 3.3 punya kemiripan dengan tabel sebelumnya yang menjelaskan jenis tagset bahasa inggris, yaitu memiliki tiga jenis kolom. Kolom nama dengan singkatan nama jenis entitas, penjelasan mengenai jenis entitas tersebut dan contoh konkret kata-kata yang dianggap jenis entitas tersebut. Untuk inspirasi tiap dataset didapatkan dari beberapa sumber. Person, organization, location adalah tagset yang paling sering digunakan. Hampir seluruh dataset umum NER menggunakan ketiga jenis ini seperti ACE 2004, ACE 2005, CoNLL 2003, MUC 7. Untuk jenis tagset Date, Time berasal dari jenis tagset yang digunakan oleh MUC 7. Dan jenis Event merupakan salah satu tagset yang bukan berasal dari tagset yang umum, namun ditemukan dalam salah satu *library* umum yaitu spaCy. Dan untuk Miscellaneous adalah jenis entitas yang ditemukan namun berada diluar ketujuh entitas yang telah ditentukan.

Bagian penjelasan ini akan mendeskripsikan contoh kasus kata-kata yang akan dianggap merupakan jenis entitas dari jenis tagset tersebut. Penentuan kata yang termasuk entitas person akan mengarah kepada nama seseorang (Basuki Tjahaja Purnama), maupun pangkat seseorang (Irjenad Letjen TNI Rudianto). Seseorang yang memiliki panggilan seperti Bapak Ahok, Saudara Rudianto akan

diketahui juga disebut entitas person. Subyek/obyek yang disebut hanya dengan jabatan seseorang dapat dianggap sebuah entitas, namun perlu adanya konteks jabatan disebut sebagai orang bukan hanya sebagai jabatan. Contohnya terdapat dua kalimat, kalimat A adalah "Itulah pekerjaan Presiden Indonesia", dan kalimat B adalah "Akan dilakukan oleh Presiden Indonesia". Kalimat A mereferensikan kata Presiden sebagai sebuah jabatan. Sedangkan untuk kalimat B, mereferensikan kata Presiden sebagai seseorang, sehingga untuk kalimat ini Presiden dapat ditentukan sebagai entitas PERSON. Namun untuk panggil yang tidak bernama seperti kata dia, perempuan/lelaki itu, pejabat tersebut, penghuni itu, panggilan-panggilan yang tidak memiliki konteks sebelumnya (seperti sebutan nama siapakan orang tersebut) siapa tidak akan ditandai sebagai entitas.

Tabel 3.3
Jenis Tagset Bahasa Indonesia

Entitas	Keterangan	Contoh
Person (PER)	Seseorang/kumpulan orang	President Obama
Organization (ORG)	Perusahaan, agensi, grup dalam struktur organisasi	Secret Service
Location (LOC)	Lokasi geografis seperti daratan, lautan, dan bentukan geologis	Red Sea
Date (DATE)	Tanggal, bulan, tahun	17 Agustus 1945, bulan Juli
Time (TIME)	Jam, menit, detik, zona waktu, kurun waktu	19:00, malam hari, Senin pagi, era Soekarno
Event (EVENT)	Kejadian, peristiwa yang dirancang untuk terjadi	Pilkada 2019, selebrasi Tahun Baru
Miscellaneous (MISC)	Entitas selain enam entitas sebelumnya	10 persen, Rp. 10.000, Undang-Undang

Definisi untuk entitas organisasi adalah sekumpulan orang yang bekerja sama untuk mencapai tujuan tertentu dan biasanya terpimpin atau terkendali. Organisasi yang mudah untuk diketahui adalah sebuah usaha atau perusahaan (Shopee, Pertamina). Adapun organisasi politik yang akan sering ditemukan dalam dataset seperti TNI, partai politik, badan legislatif. Organisasi formal yang tidak berhubungan dengan bisnis dan profesi juga dapat ditemukan seperti persekolahan,

universitas, perseroan. Organisasi sosial seperti LSM, karang taruna, PMI juga diikut sertakan sebagai entitas. Untuk kumpulan orang namun tidak ada penamaan yang jelas seperti para murid, para relawan, seluruh anak, tidak dinyatakan sebagai entitas jika tidak ada konteks lebih. Contoh entitas yang dapat digolongkan sebagai organisasi dari konteks seperti para anggota DPR.

Entitas location akan digolongkan untuk suatu daerah apa pun (daratan, perairan) yang dapat ditunjuk secara akurat dengan konteks yang memadai. Contoh umum dapat diambil dari penamaan daerah seperti kota, provinsi, negara. Bisa juga perairan atau bentuk lahan yang memiliki nama yang resmi seperti Danau Toba, Laut Sulawesi, Bukit Bintang. Bisa juga lokasi suatu gedung, tempat apa pun yang memiliki penulisan nama yang jelas seperti jalan, daerah, arah mata angin. Semisal Jawa Barat, Rumah Sakit Mata Undaan (Undaan juga dapat digolongkan sebagai location sendiri), Jl. Ngagel Jaya Tengah No. 73-77. Objek wisata dan fasilitas umum seperti Trans Studio Bandung atau Bandara Soekarno-Hatta juga termasuk dalam entitas location. Beberapa dari dokumen dataset memiliki penulisan lokasi dengan penulisan bujur dan lintang, 10.59 Lintang Selatan (LS) dan 119.42 Bujur Timur (BT) juga termasuk location. Entitas location yang ada dari dataset adalah daerah jalan yang ditentukan arah dan tujuan (KM 14 Tol Jagorawi arah menuju Cawang).

Selanjutnya entitas date cukup singkat dan jelas, yaitu penulisan tanggal baik tanggal hari, tanggal bulan, tanggal tahun. Juga penulisan nama bulan dapat dihitung sebagai entitas date. Contoh yang dapat diberikan, bulan Januari, 17/08/1945, tahun 2022, dan kombinasi penulisan tanggal lainnya. Entitas date sering dipasangkan dengan entitas time, dimana entitas time akan memiliki ambiguitas yang banyak. Ambiguitas ini sering ditemukan karena entitas time didefinisikan dengan periode waktu yang disebut. Beberapa contoh yang jelas sebagai time adalah nama hari, waktu, zona waktu, jumlah waktu, bagian hari. Juga ada contoh seperti periode, masa, dan lainnya. Semisal, hari Senin, 17:00 WIB, 1 jam 30 menit, enam hari, malam hari, tahun ajaran 2022 – 2023, era Soeharto.

Event dalam bahasa Indonesia adalah peristiwa, kejadian. Dalam tugas akhir ini, ditentukan adanya entitas ini untuk menyesuaikan dengan kebutuhan dataset

yang menyebut entitas event cukup sering. Untuk mempersingkat, entitas ini dianggap jika kata-kata yang disebut menyebut suatu nama kejadian/peristiwa yang yang sudah dirancang secara sengaja untuk terjadi. Semisal untuk kejadian pemilihan kepala daerah adalah suatu entitas event yang telah dirancang dan memiliki jadwal yang pasti tiap beberapa tahun. Bisa juga kejadian yang formal yang bukan diadakan beberapa hari seperti Pilkada, rapat paripurna atau suatu persidangan bisa juga acara yang diadakan oleh seorang organisasi seperti acara festival atau lomba.

Miscellaneous jika ditranslasi dalam bahasa Indonesia berarti aneka ragam atau bermacam-macam. Tujuan dengan adanya jenis entitas ini untuk menandakan kata-kata yang bisa termasuk named entity, namun tidak temrasuk dalam jenis yang telah di tentukan dalam named entity yang sudah ada (dari keenam jenis named entity lainnya). Beberapa contoh kata-kata yang ditemukan dalam dataset untuk named entity miscellaneous adalah entitas jenis uang (Rp. 100.000, 10 ribu rupiah, USD 100), entitas jenis dokumen/perhukuman (Pasal 107 KUHP, Undang-Undang, Peraturan Pemerintahan/PP), entitas jenis karya seperti buku novel, lukisan, lagu dan lain-lainnya (seni batik, Indonesia Raya, Ada Apa dengan Cinta), entitas jenis produk (Toyota, Suzuki APV).

3.4.2 Statistika Dataset

Dataset bahasa Indonesia pada tugas akhir ini berasal dari CNN Indonesia dengan domain berita politik. Dataset ini sebelumnya sudah pernah dipakai untuk tugas akhir Christian Nathaniel Purwanto (214116299) dan juga tugas akhir Georgia Nikita (218116685). Berjumlah 2000 dokumen, isi dari tiap dokumen merupakan berita politik Indonesia. Mayoritas berita akan menyinggung politik seperti kasus korupsi, kenaikan jabatan menteri, pelaksanaan atau kampanye Pilkada, berita kemacetan lalu lintas. Tabel 3.4 adalah tabel yang menunjukkan statistika named entity dataset pada saat tugas akhir ini dikerjakan. Statistika yang ditunjukkan adalah jumlah entitas yang bersarang dalam dataset secara keseluruhan dalam satuan persentase dengan jenis entitas/tagset yang telah dibahas pada bab sebelumnya. Detail secara keseluruhan akan dibahas di akhir bab ini.

Tabel 3.4
Spesifikasi Dataset NER Bahasa Indonesia

Statistika	Jumlah Entitas	Jumlah Entitas Bersarang	Persentase Jumlah
Person (PER)	34.720	8.488	24%
Organization (ORG)	20.857	8.627	41%
Location (LOC)	20.004	8.578	43%
Date (DATE)	5.720	3.548	62%
Time (TIME)	7.679	1.292	17%
Event (EVENT)	3.103	609	20%
Miscellaneous (MISC)	6.774	1.345	20%

3.5 Preprocessing

Pra proses dalam tugas akhir ini sempat disinggung dalam bab arsitektur sistem. Proses ini dilakukan kepada dataset tugas akhir sebelum menjadi input untuk training model nya. Pada bab arsitektur sistem, pembahasan yang diberikan lebih difokuskan kepada alur sistem dan data yang digunakan sebagai input dan output. Pada subbab ini akan dibahas lebih mendetail tiap tahap pra proses yang dilakukan pada dataset tugas akhir. Terdapat tiga sub bab yaitu cara pelabelan dataset, konversi struktur dataset dan struktur akhir dataset.

3.5.1 Struktur dan Pelabelan Dataset

Dataset tugas akhir ini terbantu dengan bantuan tugas akhir yang juga mengambil topik NLP. Beberapa tugas akhir pembantu telah disebut sebelumnya dan ada yang belum disebut. Tiap proses yang dilewatkan dataset dari tiap tugas akhir akan dibahas satu per satu. Alur besar dari penjelasan adalah sumber utama isi dataset dari website yang diambil dan juga bantuan *crawler* website. Juga format dataset yang sudah pernah dibuat, dan bentuk-bentuk akhir dataset yang bisa menjadi file input output kepada program masing-masing. Adapun library yang membantu pra proses dataset ini akan dijelaskan. Penjelasan dan sumber dataset

didapatkan dari Amelinda Tjandra Dewi (214116288)[13], Christian Nathaniel Purwanto (214116299), Georgia Nikita (218116685).

Topik/domain dari dataset adalah berita politik yang berhubungan dengan politik Indonesia, ada pun beberapa berita lainnya seperti berita lalu lintas namun mayoritas dari dataset berisi domain politik. Hal ini karena sumber berita diambil dari dua website berbeda yang menyajikan berita Indonesia yaitu CNN Indonesia[14] dan Liputan 6[15]. CNN Indonesia dan Liputan 6 tidak hanya memiliki website portal berita tetapi juga siaran televisi yang sudah mulai dari tahun 2015 dan Liputan 6 pada 2000. Kedua situs berita menyediakan berita dalam bahasa Indonesia dengan tema yang sebagian besar sejenis. Berita yang disediakan seperti berita umum, nasional, lokal, bisnis, teknologi dan hiburan, dan untuk tiap portal memiliki fokus berbeda. Untuk CNN Indonesia memiliki fokus terhadap berita nasional, internasional, politik, olahraga dan Liputan 6 memiliki beberapa fokus berita yang khusus seperti meskipun ada jenis berita bisnis, terdapat juga bagian yang fokus terhadap berita *cryptocurrency*, saham, adapun bagian berita untuk foto.

Total berita yang diambil adalah 2000 berita dari kedua website dan pembagian dibagi sesuai dengan kebutuhan masing-masing tiap tugas akhir (contohnya pada tugas akhir Amelinda pembagian training dan test bergantung pada kasus uji coba). Untuk tugas akhir ini pembagian dataset akan dibagi dalam persentase 90 untuk training dan development dan presentase 10 untuk testing. Spesifikasi detail mengenai pembagian dataset akan dibahas pada subbab berikut ini.

Pengambilan dataset dari tiap website dilakukan dalam tugas akhir Amelinda menggunakan library crawler bahasa pemrogramman Python bernama Beautiful Soup. Secara singkat, cara library tersebut mengambil data dengan library ini adalah melihat struktur HTML dari halaman web yang diminta. Alur program crawler yang dibuat adalah mengambil link berita dari website dan dikumpulkan.

---

[13] Amelinda Tjandra Dewi, Skripsi:Named Entity Recognition dan Coreference Resolution Nama Orang untuk Teks Bahasa Indonesia dengan Menggunakan Conditional Random Fields. (Surabaya:2018).
[14] CNN Indonesia, *https://www.CNNIndonesia.com*
[15] Liputan 6, *https://www.liputan6.com*

Kemudian link tersebut akan diakses dan konten dari halaman tersebut akan dicrawl dari elemen-elemennya. Contoh pengambilan judul dari elemen H1, isi dari berita adalah elemen div dengan class bernama article-content-body_item-content. Hasil *crawling* ini dimasukkan ke dalam suatu database untuk memudahkan pengumpulan data dan proses berikutnya untuk pra proses.

Input untuk tugas akhir Amelinda sama seperti dengan tugas akhir ini yaitu berupa token yang *word-level*. Karena itu hasil crawling akan berupa file ekstensi txt yang akan dilewatkan tokenisasi dan menghasilkan file ekstensi txt yang berbeda. Contoh kalimat untuk file pertama akan berisi "oleh pihak tergugat, Agus.". Sedangkan contoh hasil kalimat setelah tokenisasi pada file lainnya adalah "oleh pihak tergugat , Agus . ". File kedua (yang telah ditokenisasi) adalah file yang akan digunakan pada seluruh tugas akhir termasuk tugas akhir ini. Dan file berikutnya yang juga digunakan semua tugas akhir ada file ekstensi ann. File ini telah dijelaskan isi, struktur dan proses pembuatan file pada subbab 2.1.1 sistem arsitektur pra bagian proses. Dan file tersebut akan diubah menuju struktur data yang diperlukan model tiap tugas akhir.

Namun untuk struktur input data training model dari ketiga tugas akhir tersebut (selain tugas akhir ini), format yang dibutuhkan adalah format BIO. BIO adalah format tag dengan keterangan tag B untuk Beginning, I untuk Inside, O untuk Outside. B menyatakan bahwa token tersebut adalah awalan untuk sebuah entity, I untuk menyatakan token tersebut berada didalam suatu entity yang sudah diawali sebelumnya dengan tag B. O adalah untuk token yang tidak termasuk entity apapun. Dan seluruh tugas akhir tersebut (kecuali Georgia Nikita dan tugas akhir ini), juga akan memiliki POS tag selain BIO tag untuk setiap kata. Isi dari struktur data tugas akhir Christian dan Amelinda dapat dilihat sebagai berikut:

0	Menteri	JJ	B-JOB	O	-1
1	Luar	JJ	I -JOB	O	-1
2	Negeri	NN	I -JOB	O	-1
3	Iran	NNP	B-PER	Corefer-to	0
4	Javad	NNP	I-PER	O	-1

Berikut adalah isi dari file dengan ekstensi .bio untuk input training model. Kolom pertama menyatakan kode untuk tiap kata, kolom kedua adalah isi dari kata

yang dilabelkan, kolom kedua sebagai isi dari POS tag, berikutnya adalah tag named entity dalam format IOB, dan dua kolom terakhir digunakan untuk mengetahui informasi *coreference* yang ada. Apabila kata tersebut memiliki nilai coreference terhadap kata lain. Kolom kedua terakhir adalah jenis relasi dan kolom terakhir ada kode kata yang kata tersebut memiliki coreference. Coreference dapat ditemukan dalam dataset ini karena tujuan tugas akhir tersebut meneliti kombinasi NER dan coreference.

Untuk tugas akhir ini, memiliki struktur akhir dataset (data input) mengikuti struktur yang telah diberikan dari penelitian metode Sequence-to-Set Network. File input akan berupan ekstensi .json, satu file adalah sekumpulan JSON objek dengan atribut yang sudah ditentukan. Sebelum menjelaskan isi dari dataset perlu diketahui bahwa dokumen yang disebut adalah satu teks berita dengan berbagai kalimat. File input untuk tugas akhir ini hanya tunggal karena seluruh dokumen akan digabung menjadi satu. Struktur secara keseluruhan untuk satu JSON objek dapat dilihat pada Gambar 3.10.


```
{
    "tokens": [
        "Menurut","Ahok",",","perombakan","jabatan","tersebut",
        "mungkin","akan","dilakukan","gubernur","baru",",",
        "untuk","masa","pemerintahan","2017-2022","."
    ],"entities": [
        {"start": 1, "end": 2, "type": "PER"}, {"start": 13, "end": 16, "type": "TIME"},
        {"start": 14, "end": 15, "type": "ORG"}, {"start": 15, "end": 16, "type": "DATE"},
        {"start": 15, "end": 16, "type": "DATE"}
    ],"relations": {},"org_id": "2934116",
    "pos": [
        "<UNK>","<UNK>","<UNK>","<UNK>","<UNK>","<UNK>",
        "<UNK>","<UNK>","<UNK>","<UNK>","<UNK>","<UNK>",
        "<UNK>","<UNK>","<UNK>","<UNK>","<UNK>"
    ],
    "ltokens": [
        "Gubernur","DKI","Jakarta","Basuki","Tjahaja","Purnama",
        "atau","Ahok","mengaku","tidak","mengetahui","rencana",
        "pergantian","Satuan","Kerja","Perangkat","Daerah","(",
        "SKPD",")","."
    ],
    "rtokens": [
        "\"","Siapa","itu","yang","usul","Tunggu",
        "saja","saat","ganti","gubernur","baru",",",
        "\"","kata","Ahok","di","Balai","Kota",
        "Jakarta",",","Kamis","(","27/4/2017",")","."
    ]
}
```


Gambar 3.10
Contoh Struktur Dataset

Terdapat tujuh atribut yang diperlukan untuk input training model metode ini. Setiap JSON objek dari dalam dataset harus memiliki ketujuh atribut, meskipun

beberapa atribut tidak akan digunakan sebagai parameter training dalam penelitian tugas akhir ini. Nama-nama atribut secara urut adalah tokens, entities, relations, org_id, pos, ltokens, rtokens. Tokens adalah kalimat utama dari JSON objek tersebut. Isi dari tokens diambil dari dokumen yang dipecahkan berdasarkan kalimatnya, kemudian kalimat-kalimat tersebut akan dipecah lagi untuk tiap kata. Karena itu isi dari tokens adalah *array of word-level token* (word-level artinya dipecahkan berdasarkan kata-kata). Entities cukup jelas yaitu array dari named entity untuk kalimat itu saja. Penyimpanan named entity nya menggunakan JSON objek juga dengan tiga atribut start, end dan type. Start dan end adalah indeks, namun berbeda dengan BRAT, indeks ini akan mengikuti bentuk tokens yaitu menjadi indeks untuk kata keberapa (word-level bukan *character-level*). Dan type adalah jenis named entity untuk kata tersebut.

Kedua atribut berikutnya, relations dan org_id, tidak akan digunakan sebagai input ke dalam training model, karena untuk relations digunakan untuk coreference yang bisa dijadikan tambahan informasi untuk model namun tidak digunakan dalam penelitian tugas akhir ini. Untuk org_id akan berisi kode untuk tiap asal dokumen untuk kalimat tersebut, sebagai bantuan dokumentasi saja. Ltokens dan rtokens dapat dijelaskan bersamaan karena memiliki kegunaan yang sama. Isi dari kedua atribut tersebut sama dengan atribut tokens, tetapi ltokens akan mengambil kalimat sebelum kalimat utama dan rtokens akan mengambil kalimat setelah kalimat utama. Huruf l dalam ltokens adalah *left*/kiri, huruf r dalam rtokens adalah *right*/kanan, mereferensikan kanan dan kiri kalimat utama.

3.5.2 Konversi dan Statistika Dataset

Dari subbab struktur dan pelabelan dataset, diketahui bahwa tidak semua metode menerima struktur data yang sama, karena itu dari struktur data alat pelabelan menuju model perlu konversi. Tugas akhir ini telah menyediakan konversi dataset dari file BRAT menjadi file JSON dataset. Beberapa bagian penting untuk konversi dataset akan dijelaskan dalam subbab ini, perlu diketahui bahasa pemrogramman untuk konversi dataset menggunakan Python. Yang akan dijelaskan adalah segmen program untuk preprocessing file txt berisi dokumen teks

berita yang sudah ditokenisasi, segmen program preprocessing file ann berisi tag named entity untuk tiap dokumen, juga preprocssing untuk penyesuaian indeks BRAT dengan indeks word-level.

Segmen Program 3.1 Preprocessing TXT File

```
# Preprocessing TXT File
batas_files = 2000
ctr_files = 0
for filename in filenames :
  ctr_files = ctr_files + 1
  if (ctr_files <= batas_files) :
    with open(str(filename)+".txt", 'r') as f:
      t_doc = []
      temp = f.read().split(' \n')
      first_sen= True
      last = 0
      for idx, t in enumerate(temp) :
        t_first_idx = -1
        t_tokens = t.split(' ')

        if first_sen == True :
          t_first_idx = 0
          first_sen = False
        else :
          t_first_idx = last + 2
        t_last_idx = len(t) + t_first_idx
        last = t_last_idx
        t_ltokens = []
        t_rtokens = []
        if idx-1 >= 0 :
          t_ltokens = temp[idx-1].split(' ')
        if idx+1 <= len(temp)-1 :
          t_rtokens = temp[idx+1].split(' ')
        t_pos = ['<UNK>' for i in t_tokens]
        t_obj = {
            "tokens" : t_tokens,
            "first_idx" :  t_first_idx,
            "last_idx" :  t_last_idx,
            "entities": [],
            "pos": t_pos,
            "ltokens": t_ltokens,
            "rtokens": t_rtokens
        }
        t_doc.append(t_obj)

      documents.append({
          "id_doc" : str(filename),
          "t_doc" : t_doc
      })
len(documents)
```

Input Segmen Program 3.1 adalah txt file dokumen berita yang telah ditokenisasi tiap kata dan juga tanda baca (petik, koma, titik, garis miring). Output dari segmen program ini adalah variabel *documents* yang merupakan kumpulan objek oleh tiap dokumen dengan atribut nama dan detail dokumen. Baris 13 – 22 adalah perintah program yang berperan untuk menghitung indeks pertama dan indeks terakhir untuk tiap baris di dokumen tersebut. Contohnya untuk kalimat pertama memiliki indeks awal dan akhir yaitu 0 dan 23, sedangkan mungkin untuk kalimat kedua memiliki indeks awal dan akhir yaitu 25 dan 50. Perlu diperhatikan dari kalimat pertama dan kedua terdapat jarak indeks sebanyak dua, hal ini karena indeks dari BRAT juga menghitung karakter spesial yaitu '\n'. Dengan ini indeks tiap awal akan ditambah sebanyak dua agar lebih tepat. Indeks awal dan akhir tiap kalimat akan dijelaskan perannya pada segmen program pra proses ANN file.

Untuk baris 23 – 28 akan melakukan pra proses untuk mendapatkan nilai ltokens dan rtokens. Baris selanjutnya mengisi variabel untuk POS tag, namun karena tugas akhir ini tidak menggunakan POS tag maka variabel ini akan diisi dengan tokens <UNK> diketahui sebagai token untuk kata yang tidak dikenal. Namun ini tidak akan mempengaruhi penelitian karena parameter untuk mengaktifkan penggunaan POS embedding dimatikan (nilai parameter diatur menjadi *false*). Dan baris 30 – 38 adalah deklarasi variabel objek berisi detail yang dibutuhkan untuk struktur dataset akhir nanti.

Segmen Program 3.2 Preprocessing ANN File

```
01:      for idx, t in enumerate(temp) :
02:        t_splitted_tab = t.split('\t')
03:        if len(t_splitted_tab) > 1 :
04:          is_error = False
05:
06:          try:
07:            t_splitted_space = t_splitted_tab[1].split(' ')
08:            t_splitted_space[1] = int(t_splitted_space[1])
09:            t_splitted_space[2] = int(t_splitted_space[2])
10:          except:
11:            is_error = True
12:            print('FILENAME ERROR : ', filename,
13:            'KODE ERROR : ', t_splitted_tab[0],
14:            'kalimat : ', t_splitted_space)
```

Segmen Program 3.2 dijalankan setelah mendapat isi dari dokumen, kemudian label data yang telah dilakukan akan dilewatkan pra proses juga agar dapat dimasukkan ke dalam variabel program. Perintah yang akan dipakai adalah *split* karena penulisan format dalam file ann sangat templat. Bentuk dan format telah dijelaskan di bab ketiga mengenai arsitektur sistem bagian pra proses. Secara singkat, baris 2 akan menerima satu baris dari file ann kemudian dipisahkan berdasarkan karakter spesial tab (\t), hasil dari pemisahan tersebut adalah mendapatkan kode label, indeks label dan jenisnya, yang terakhir adalah kata-kata yang dilabelkan. Baris 6 – 14 berusaha melakukan perintah split karena indeks label dan jenisnya bergabung menjadi satu dengan pemisahan spasi (‘ ‘). Apabila terjadi kegagalan artinya ada penulisan di file tersebut yang tidak sesuai format BRAT, sehingga file tersebut tidak akan dimasukkan ke dalam dataset dan akan dicetak nama file untuk mempermudah preprocess secara manual.

Segmen Program 3.3 Preprocess Indeks ANN

```
01:   def convert_index(entities_coba, tokens):
02:     entities_converted = []
03:     str_coba = " ".join(tokens)
04:
05:     for idx_e, e in enumerate(entities_coba) :
06:       start_idx = e['start']
07:       end_idx = e['end']
08:       ctr = -1
09:       sum = 0
10:
11:       start_entity = -1
12:       end_entity = -1
13:       for kata in str_coba.split(" "):
14:         ctr = ctr + 1
15:         sum = sum + len(kata) + 1
16:
17:         if sum > start_idx and start_entity == -1:
18:           start_entity = ctr
19:         if sum > end_idx and end_entity == -1:
20:           end_entity = ctr
21:           temp = {
22:               'start' : start_entity,
23:               'end' : end_entity+1,
24:               'type' : e['tag']
25:           }
26:           entities_converted.append(temp)
27:           # print(temp)
28:           break
29:
30:     return entities_converted
```

Segmen program terakhir nomor Segmen Program 3.3, digunakan untuk mengubah indeks awal dan akhir yang sesuai alat pelabelan yaitu BRAT menjadi indeks yang esuai dengan struktur data akhir untuk training model. Indeks yang digunakan BRAT menggunakan format indeks character-level (indeks melihat huruf keberapa) dan indeks dimulai dari angka nol untuk satu dokumen, bukan untuk tiap kalimat. Untuk Sequence-to-Set Model menggunajan format indeks word-level (indeks melihat kata keberapa) dimana tiap kalimat mulai dari 0. Cara program berjalan pada inti program yaitu baris 14 – 25. Dari indeks yang telah diberikan, tiap kalimat yang telah dibuat dari tokens yang dimiliki akan dihitung panjangnya. Penghitugan panjang dari kalimat akan dilakukan per kata dan apabila panjang tersebut lebih dikit daripada indeks label saat itu maka akan dicatat sebagai indeks pertama atau terakhir. Jika kedua indeks ditemukan maka langsung akan dibuatkan objek baru dengan konversi tersebut.

Tabel 3.5
Statistika Dataset Tugas Akhir

Statistika	Jumlah
Kalimat	32.355
Kalimat dengan entitas bersarang	12.147
Rata-rata panjang kalimat	19,04132
Seluruh entitas	98.857
Seluruh entitas bersarang	32.487
Persentase entitas (%)	32%

Berikut adalah statistika dari dataset yang telah melewatkan pelabelan terbaru. Tabel 3.5 menampilkan beberapa detail dataset yang diambil dari tabel spesifikasi dataset oleh penelitian tugas akhir ini. Seluruh kalimat (kalimat yang memiliki atau tidak memiliki pelabelan terhitung semua) dari dataset telah dihitung dengan jumlah sebanyak 32.355, dan untuk kalimat yang memiliki entitas bersarang berjumlah 12.341 sebanyak 38% kalimat yang mengandung entitas bersarang. Sebagai informasi tambahan, panjang kalimat secara keseluruhan berkisaran 19. Dan keterangan mengenai entitas yang berada didalam dataset terdapat 98.857

entitas (termasuk entitas bersarang). Dan didalam jumlah seluruh entitas, yang mengandung entitas bernama sebesar 20.745, ini berarti 21% dari seluruh entitas memiliki entitas bersarang.