TUGAS AKHIR – IF184802

PENENTUAN REKOMENDASI TUJUAN WISATA DI INDONESIA DARI DATA TIDAK TERSTRUKTUR DENGAN *NAMED ENTITY RECOGNITION* , METODE CLUSTERING *K-MEANS* DAN *K-NEAREST NEIGHBOR*

DENISE SONIA RAHMADINA
NRP 05111640000177

Dosen Pembimbing
Abdul Munif S.Kom, M.Sc.
Nurul Fajrin Ariyani, S.Kom, M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]

TUGAS AKHIR – IF184802

PENENTUAN REKOMENDASI TUJUAN WISATA DI INDONESIA DARI DATA TIDAK TERSTRUKTUR DENGAN *NAMED ENTITY RECOGNITION*, METODE CLUSTERING *K-MEANS* DAN *K-NEAREST NEIGHBOR*

DENISE SONIA RAHMADINA
NRP 05111640000177

Dosen Pembimbing
Abdul Munif S.Kom, M.Sc.
Nurul Fajrin Ariyani, S.Kom, M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]

FINAL PROJECT – IF184802

DETERMINATION OF TOURIST DESTINATION RECOMMENDATIONS IN INDONESIA FROM UNSTRUCTURED DATA WITH NAMED ENTITY RECOGNITION, K-MEANS CLUSTERING METHOD AND K-NEAREST NEIGHBOR

DENISE SONIA RAHMADINA
NRP 05111640000177

Advisor
Abdul Munif S.Kom, M.Sc.
Nurul Fajrin Ariyani, S.Kom, M.Sc.

INFORMATICS ENGINEERING DEPARTMENT
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PENENTUAN REKOMENDASI TUJUAN WISATA DI INDONESIA DARI DATA TIDAK TERSTRUKTUR DENGAN *NAMED ENTITY RECOGNITION* , METODE CLUSTERING *K-MEANS* DAN *K-NEAREST NEIGHBOR*

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Manajemen Informasi
Program Studi S-1 Teknik Informatika
Departemen Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh:
DENISE SONIA RAHMADINA
NRP: 05111640000177

1. Abdul Munif., S.Kom., M.Sc. .....
   NIP. 19860823 201504 1 004 (Pembimbing 1)

2. Nurul Fajrin Ariyani, S.Kom., M.Sc. .....
   NIP. 19860722 201504 2 003 (Pembimbing 2)

SURABAYA

LEMBAR PENGESAHAN

PENENTUAN REKOMENDASI TUJUAN WISATA DI INDONESIA DARI DATA TIDAK TERSTRUKTUR DENGAN *NAMED ENTITY RECOGNITION*, METODE CLUSTERING *K-MEANS* DAN *K-NEAREST NEIGHBOR*

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Manajemen Informasi
Program Studi S-1 Teknik Informatika
Departemen Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh:
DENISE SONIA RAHMADINA
NRP: 05111640000017

1. Abdul Munif., S.Kom., M.Sc. .....
   NIP. 19860823 201504 1 004 (Pembimbing 1)

2. Nurul Fajrin Ariyani, S.Kom., M.Sc. .....
   NIP. 19860722 201504 2 003 (Pembimbing 2)

SURABAYA
JANUARI 2020

[Halaman ini sengaja dikosongkan]

PENENTUAN REKOMENDASI TUJUAN WISATA DI INDONESIA DARI DATA TIDAK TERSTRUKTUR DENGAN *NAMED ENTITY RECOGNITION* , METODE CLUSTERING *K-MEANS* DAN *K-NEAREST NEIGHBOR*

Nama Mahasiswa : Denise Sonia Rahmadina
NRP : 05111640000177
Departemen : Informatika ITS
Dosen Pembimbing 1 : Abdul Munif S.Kom, M.Sc.Eng
Dosen Pembimbing 2 : Nurul Fajrin Ariyani, S.Kom, M.Sc.

ABSTRAK

Pariwisata merupakan suatu varian yang kompleks mencakup serangkaian jenis operasi (transaksi, aktivitas atau peristiwa di pasar pariwisata) seperti pencarian web, kunjungan halaman web, online pemesanan & pembelian, dll. Sehingga menghasilkan data dengan jumlah yang signifikan untuk dilakukan proses pengolahan data untuk membantu para wisatawan dalam menentukan tujuan wisata yang diinginkan.

Dalam tugas akhir ini, dilakukan pembangunan model dengan Named Entity Recognition (NER), Part-of-speech (POS) Tagger, dan Rule Based Matching untuk mendeteksi entitas untuk membantu menentukan tujuan rekomendasi objek wisata di Indonesia. Selanjutnya pada tugas akhir ini, dilakukan clustering dan klasifikasi data dengan metode K-Means dan K-Nearest Neighbor untuk membagi tujuan wisata ke dalam kategori dan sesuai lokasi.

Dari hasil evaluasi, didapatkan bahwa model Named Entity Recognition (NER) memiliki akurasi 99,7% dalam melabeli

data, hasil clustering dengan K-Means menghasilkan 10 cluster data dengan akurasi 85% dan hasil klasifikasi dengan K-Nearest Neighbor dengan akurasi 90,1%.

Kata kunci: Pariwisata, Named Entity Recognition (NER), clustering, klasifikasi.

DETERMINATION OF TOURIST DESTINATION RECOMMENDATIONS IN INDONESIA FROM UNSTRUCTURED DATA WITH NAMED ENTITY RECOGNITION , K-MEANS CLUSTERING METHOD AND K-NEAREST NEIGHBOR

Name : Denise Sonia Rahmadina
NRP : 05111640000177
Department : Informatics FTIK-ITS
Supervisor I : Abdul Munif S.Kom, M.Sc.Eng
Supervisor II : Nurul Fajrin Ariyani, S.Kom, M.Sc.

ABSTRACT

Tourism is a complex variant that includes a series of types of operations (transactions, activities or events in the tourism market) such as web search, web page visits, online ordering & purchasing, etc. So as to produce a significant amount of data to do the data processing to help tourists in determining the desired tourist destination.

In this final project, a model construction with Named Entity Recognition (NER), Part-of-speech (POS) Tagger, and Rule Based Matching is carried out to detect entities to help determine the destination of tourist attraction recommendations in Indonesia. Furthermore, in this final project, clustering and data classification is carried out using the K-Means and K-Nearest Neighbor methods to divide tourist destinations into categories and according to location.

From the evaluation results, it was found that the Named Entity Recognition (NER) model has an accuracy of 99.7% in labeling the data, the results of clustering with K-Means produce 10 data clusters with an accuracy of 85% and the results of the classification with K-Nearest Neighbor with 90.1 %.

Keywords: tourism, Named Entity Recognition (NER), clustering, classification

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah Subhanahu Wa Ta'ala karena atas karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

"**PENENTUAN REKOMENDASI TUJUAN WISATA DI INDONESIA DARI DATA TIDAK TERSTRUKTUR DENGAN *NAMED ENTITY RECOGNITION* , METODE CLUSTERING *K-MEANS* DAN *K-NEAREST NEIGHBOR***"

Terselesaikannya Tugas Akhir ini tidak terlepas dari bantuan dan dukungan dari banyak pihak, Oleh karena itu melalui lembar ini penulis ingin mengucapkan terima kasih dan penghormatan kepada :

1. Allah SWT Yang Maha Kuasa, karena limpahan rahmat dan karunia-Nya penulis dapat menjalankan perkuliahan di Departemen Informatika Institut Teknologi Sepuluh Nopember dan dapat menyelesaikan Tugas Akhir guna memenuhi syarat kelulusan sebagai Sarjana.
2. Kedua orangtua penulis Papi, Mami, adik penulis Andoni, Andriyo dan Junbobi yang telah memberikan dukungan doa, dan bentuk apapun kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Bapak Abdul Munif, S.Kom, M.Sc. dan Ibu Nurul Fajrin Ariyani, S.Kom., M.Sc. selaku pembimbing I dan II yang telah membimbing, memberi nasihat, serta mengorbankan waktu dan tenaga untuk membimbing penulis dalam menyelesaikan Tugas Akhir ini.
4. Dr. Eng. Darlis Herumurti, S.Kom., M.Kom. selaku Ketua Departemen Informatika ITS dan segenap dosen dan karyawan Departemen Informatika ITS yang telah memberikan ilmu dan pengalaman kepada penulis selama menjalani masa kuliah di Informatika ITS.

5. Alhamdullilah Teman & Marvell University yang selalu ada menemani, memberikan support sepenuhnya kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
6. Devyta, Astrid, Andra, Mimi, Zane, Adela, Ario dan Rafdi yang selalu ada memberikan nasihat, pesan, menemani dan memberikan support utamanya kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
7. Teman-Teman Multichat ber-5 (Power Rangers), yaitu Dewi, Almas, Diana, dan Isye sebagai yang telah mengisi hari-hari penulis di TC.
8. Fadilla, Nila, Akbar sebagai teman seperjuangan Tugas Akhir yang selalu memberi semangat dan bantuannya dalam pengerjaan Tugas Akhir ini.
9. Teman-teman satu angkatan Informatika ITS 2016 yang selalu menyemangati satu sama lain.
10. Pihak-pihak lain yang tidak bisa penulis sebutkan satu per satu.

Penulis memohon maaf jika terdapat kesalahan maupun kekurangan dalam Tugas Akhir. Oleh karena itu dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan untuk kedepannya. Semoga laporan Tugas Akhir ini dapat berguna bagi pembaca.

Surabaya, 16 Januari 2020

Denise Sonia Rahmadina

DAFTAR ISI

DAFTAR GAMBAR

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

DAFTAR KODE SUMBER

[Halaman ini sengaja dikosongkan]

BAB I
PENDAHULUAN

1.1 Latar Belakang

Saat ini informasi yang disimpan meningkat pesat dari hari ke hari dan mengekstraksi informasi yang berguna dari volume data yang berkembang pesat. Banyak yang menyebut arus data yang sangat besar ke dalam perusahaan saat ini sebagai "Big Data" – terabyte dan gigabytes byte yang membanjiri banyak infrastruktur teknologi informasi perusahaan Sebagian besar pertumbuhan eksplosif ini terdiri dari informasi yang tidak terstruktur, yang menciptakan jenis-jenis tantangan baru dalam hal praktik tata kelola, manajemen dan keamanan. Salah satunya ada dalam kasus pariwisata [1].

Pariwisata merupakan suatu varian yang kompleks mencakup serangkaian jenis operasi (transaksi, aktivitas atau peristiwa di pasar pariwisata) seperti pencarian web, kunjungan halaman web, online pemesanan & pembelian, dll. Sehingga menghasilkan data dengan jumlah yang signifikan untuk dilakukan proses pengolahan data untuk meningkatkan pemasaran pariwisata [1]. Salah satunya adalah dengan menentukan rekomendasi tujuan wisata di Indonesia bagi para wisatawan.

Penentuan rekomendasi tujuan wisata di Indonesia dapat dilakukan dengan mengolah data tidak terstuktur yang tersebar di Internet berupa artikel atau berita tentang tujuan-tujuan wisata yang ada di Indonesia. Tujuannya adalah untuk memudahkan para wisatawatan di Indonesia untuk menentukan tujuan wisata mereka di daerah di Indonesia. Karena pada saat ini, data tujuan wisata yang tersebar di Internet masih dalam bentuk yang tidak terstruktur seperti artikel atau berita, maka data tidak terstruktur tersebut harus diolah dan diubah menjadi informasi yang lebih terstruktur dan dapat dengan mudah di konsumsi oleh para wisatawan.

Pengolahan data tidak terstruktur dilakukan dengan mengekstrak entitas-entitas yang dianggap penting untuk menentukan tujuan rekomendasi seperti lokasi dan kategori wisata. Ekstraksi entitas ini dilakukan dapat dilakukan dengan metode *Named Entity Recognition (NER)* yang akan mengekstraksi entitas-entitas yang selanjutnya dapat di lakukan penentuan rekomendasi tujaun wisata sesuai entitas yang telah di ekstraksi dengan metode *clustering K-Means* dan klasifikasi *K Nearest Neighbor (KNN)*.

Diharapkan dengan diolahnya data tidak terstruktur pada studi kasus Pariwisata di Indonesia dapat memudahkan penyedia wisata dan wisatawan untuk menentukan tempat wisata yang tepat dan sesuai dengan efektif dan efisien.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana pengumpulan, *tagging*, dan pengelompokkan data tidak terstruktur pada studi kasus Pariwisata di Indonesia?
2. Bagaimana merancang model dengan metode *Named Entity Recognition (NER)* pada studi kasus Pariwisata di Indonesia?
3. Bagaimana menentukan rekomendasi tujuan wisata dengan model yang sudah dirancang dengan algoritma *K-Means* dan *K-Nearest Neighbour (KNN)* pada studi kasus Pariwisata di Indonesia?

1.3 Batasan Permasalahan

Batasan masalah pada tugas akhir ini, sebagai berikut:

1. Dataset yang dipakai bersumber dari :
   i. Data Teks Online

Data teks berupa artikel atau berita tentang tujuan wisata di Indonesia.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah:

a) Dapat melakukan pengumpulan, *tagging*, dan pengelompokan data tidak terstruktur pada studi kasus Pariwisata di Indonesia.
b) Dapat merancang model ekstraksi entitas dengan metode *Named Entity Recognition (NER)* pada studi kasus Pariwisata di Indonesia.
c) Dapat menentukan rekomendasi tujuan wisata dengan model yang sudah dirancang dengan algoritme *K-Means* dan *K-Nearest Neighbour (KNN)* pada studi kasus Pariwisata di Indonesia.

1.5 Manfaat

Manfaat pembuatan tugas akhir ini diharapkan dapat memberikan rekomendasi tempat wisata yang terstruktur secara efektif dan efisien untuk para wisatawan di Indonesia.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. **Studi literatur**

   Pada tahap ini, akan dipelajari beberapa referensi yang diperlukan untuk pengerjaan tugas akhir, yaitu Pariwisata, *Web Crawler*, *Data Mining*, *Natural*

Language Processing (NLP), *Part-Of-Speech (POS Tagging), NER (Named Entity Recognition), Rule Based Matching, spaCy, Prodigy, Flask, K-Means, K-Nearest Neighbor*

2. **Pengambilan dan pra-proses data**

Pada tahap ini akan dilakukan proses pengambilan data dengan menggunakan *Web Crawler*, data berasal dari data user dan data operasi berbentuk artikel, berita dan deskripsi pada studi kasus Pariwisata pada media variabel, platform berita, dan hasil penelusuran web. Selanjutnya dilakukan langlah pra-proses data yaitu pembersihan data, tokenization, stop-word removal, stemming.

3. **Proses Pembuatan Model**

Pada tahap ini dilakukan proses pembuatan model *Part-Of-Speech (POS)* Tagger Bahasa Indonesia, pembuatan model *Named Entity Recognition (NER)* dan pembuatan model rekomendasi wisata dengan metode clustering dan klasifikasi *K-Means* dan *K-Nearest Neighbor.*

4. **Pengujian dan evaluasi**

Pada tahap ini dilakukan uji dan evaluasi data dari model rekomendasi wisata yang telah dibuat. Uji dan evaluasi data yang dilakukan adalah sebagai berikut :

1. Akurasi data dari model rekomendasi wisata yang telah dibuat.
2. Pengujian model dengan melakukan input pada aplikasi uji coba.

5. **Penyusunan buku Tugas Akhir**

   Pada tahap ini dilakukan proses dokumentasi dan pembuatan laporan dari seluruh konsep, tinjauan pustaka, metode, implementasi, proses yang telah dilakukan, pengujian, evaluasi dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

1.1 Sistematika Penulisan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan dan manfaat pembuatan tugas akhir, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan beberapa pustaka-pustaka yang dijadikan penunjang dan berhubungan dengan pokok pembahasan yang mendasari pembuatan tugas akhir.

BAB III DESAIN DAN PERANCANGAN SISTEM

Bab ini membahas mengenai desain dan perancangan sistem yang akan dibangun.

BAB IV IMPLEMENTASI SISTEM

Bab ini membahas mengenai bagaimana implementasi sistem dari desain yang sudah dirancang.

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dari metode yang ditawarkan dalam tugas akhir untuk mengetahui kesesuaian metode dengan data yang ada.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil pengujian yang telah dilakukan. Bab ini juga membahas saran-saran untuk pengembangan sistem lebih lanjut.

DAFTAR PUSTAKA

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

LAMPIRAN

Merupakan bab tambahan yang berisi data atau daftar istilah yang penting pada tugas akhir ini.

BAB II
TINJAUAN PUSTAKA

Bab ini membahas pustaka/teori-teori yang menjadi dasar dalam pembuatan tugas akhir.

2.1 Pariwisata

Secara umum pariwisata merupakan suatu perjalanan yang dilakukan seseorang untuk sementara waktu yang diselenggarakan dari suatu tempat ke tempat yang lain dengan meninggalkan tempat semula dan dengan suatu perencanaan atau bukan maksud untuk mencari nafkah di tempat yang dikunjunginya, tetapi semata-mata untuk menikmati kegiatan pertamasyaan atau rekreasi untuk memenuhi keinginan yang beraneka ragam [2].

Pariwisata adalah perjalanan dari suatu tempat ketempat lain, bersifat sementara, dilakukan perorangan atau kelompok, sebagai usaha mencari keseimbangan atau keserasian dan kebahagian dengan lingkungan dalam dimensi sosial, budaya, alam dan ilmu. Dorongan kepergiannya adalah karena berbagai kepentingan baik karena kepentingan ekonomi, sosial, budaya, politik, agama, kesehatan maupun kepentingan lain [2].

Seorang atau sekelompok orang yang melakukan perjalanan wisata biasanya sekedar untuk *refreshing* dan untuk berjalan-jalan. Selain itu ada yang melakukan perjalanan wisata dengan beberapa kegiatan berupa urusan bisnis ke suatu daerah tertentu. Adapun beberapa jenis pariwisata berdasarkan tujuan seseorang atau sekelompok orang yang melakukan perjalanan wisata yakni sebagai berikut :

1. Wisata Kuliner
   Wisata ini tidak hanya untuk mengenyangkan dan memanjakan perut dengan aneka ragam masakan khas dari berbagai daerah tujuan wisata, tetapi juga mendapatkan pengalaman yang menarik juga menjadi motivasinya.
2. Wisata Olahraga

Wisata ini memadukan kegiatan olahraga dengan kegiatan wisata. Kegiatan dalam wisata ini dapat berupa kegiatan olahraga yang aktif mengharuskan wisatawan melakukan gerakan olah tubuh langsung.

3. Wisata Komersial
   Wisatawan yang melakukan perjalanan untuk mengunjungi pameran-pameran dan pekan raya yang bersifat komersial seperti pameran indutsri, pameran dagang dan sebagainya.
4. Wisata Bahari
   Perjalanan yang banyak dikaitkan dengan olahra air seperti danau, pantai, air laut.
5. Wisata Industri
   Perjalanan yang dilakukan oleh rombongan mahasiswa atau pelajar yang pergi ke suatu tempat perindustrian dengan maksud dan tujuan untuk mengadakan penelitian.
6. Wisata Cagar Alam
   Jenis wisata yang banyak diselenggarakan oleh agen atau biro perjalanan yang mengkhususkan usaha-usaha dengan mengatur wisata ke tempat atau cagar alam, taman lindung, pegunungan, hutan daerah dan sebagainya yang kelestariannya dilindungi oleh Undang – Undang.

Pariwisata akan menjadi *domain* yang digunakan pada tugas akhir ini dan semua *dataset* yang digunakan berupa artikel dan berita merupakan berasal dari domain tujuan pariwisata yang ada di Indonesia.

2.2 Web Crawler

Web Crawler adalah suatu program atau *script* otomatis yang simple, yang dengan metode tertentu melakukan scan atau *"crawl*" ke semua halaman-halaman Internet untuk membuat

index dari data yang dicarinya. Nama lain untuk *web crawl* adalah *web spider, web robot, bot, crawl dan automatic indexer* [3].

Web crawling dapat digunakan untuk beragam tujuan. Penggunaan yang paling umum adalah yang terkait dengan search engine. *Search engine* menggunakan *web crawl* untuk mengumpulkan informasi mengenai apa yang ada di halaman-halaman web. Tujuan utamanya adalah mengumpulkan data sehingga ketika pengguna Internet mengetikkan kata pencarian di komputernya, search engine dapat dengan segera menampilkan *web site* yang relevan [3] . Cara kerja *web crawler* dapat disimpulkan sebagai berikut :

1. Mesin pencari web bekerja dengan cara menyimpan informasi tentang banyak halaman web, yang diambil langsung dari WWW. Halaman-halaman ini diambil dengan *web crawler — browser* web otomatis yang mengikuti setiap pranala yang dilihatnya. Isi setiap halaman lalu dianalisis untuk menentukan cara mengindeksnya (misalnya, kata-kata diambil dari judul, subjudul, atau field khusus yang disebut meta tag). Data tentang halaman web disimpan dalam sebuah database indeks untuk digunakan dalam pencarian selanjutnya.
2. Mesin pencari juga menyimpan dan memberikan informasi hasil pencarian berupa pranala yang merujuk pada *file*, seperti *file* audio, *file* video, gambar, foto dan sebagainya.
3. Ketika seorang pengguna mengunjungi mesin pencari dan memasukkan query, biasanya dengan memasukkan kata kunci, mesin mencari indeks dan memberikan daftar halaman web yang paling sesuai dengan kriterianya.

Web Crawler pada tugas akhir ini digunakan untuk melakukan tahap pengumpulan dan pengambilan data. *Dataset* yang digunakan berasal dari *website* yang mem-*publish* artikel

atau berita tentang tujuan wisata di Indonesia yang mana akan diambil dengan metode crawling dengan *Web Crawler*.

2.3 *Natural Language Processing (NLP)*

Natural Language Processing (Pengolahan Bahasa Alami) adalah pembuatan program yang memiliki kemampuan untuk memahami bahasa manusia. Pada prinsipnya bahasa alami adalah suatu bentuk representasi dari suatu pesan yang ingin dikomunikasikan antar manusia [4].

(NLP) adalah upaya untuk mengekstrak lebih jauh representasi dari suatu teks bebas. Hal ini dapat dimasukkan secara kasar seperti mencari siapa melakukan apa kepada siapa, kapan, di mana, bagaimana dan mengapa. *NLP* biasanya membuat penggunaan konsep-konsep linguistic seperti kata benda, kata kerja, kata sifat, dan lainnya dan struktur gramatikal (baik direpresentasikan sebagai ungkapan-ungkapan seperti frase nomina atau frase preposisional, atau hubungan ketergantungan seperti subjek dari- atau objek-dari) [4].

Natural Language Processing digunakan pada tugas akhir ini untuk membantu mengolah *Dataset* yang semula tidak terstruktur menjadi lebih terstruktur.

2.4 Part-Of-Speech (POS) Tagging

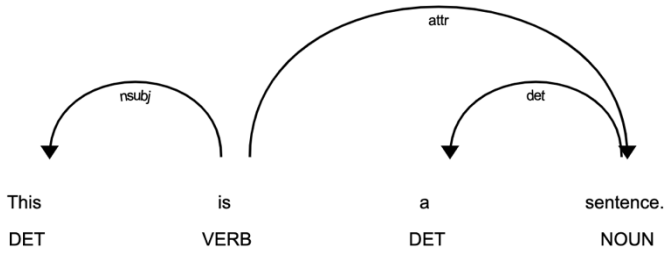
POS (Part-Of-Speech) Tagging adalah proses memberi label pada setiap kata dalam kalimat dengan *POS* atau *tag* yang sesuai untuk kata tersebut. *Tagging* dapat dimanfaatkan pada aplikasi bahasa alami lainnya, seperti sistem tanya jawab, informasi ekstraksi. Beberapa penggunaan *POS-Tagging* adalah untuk menghapus perbedaan yang tidak relevan, menghapus ambiguitas, membantu *stemming* dan membantu pencarian kata benda [5]. Terdapat 36 kategori *POS-Tag* yang termuat dalam Penn Treebank. Dapat dilihat pada Tabel 2.1

Tabel 2.1 POS Tag

No.	*Tag*	*Deskripsi*
1.	CC	*Coordinating conjunction*
2.	CD	*Cardinal number*
3.	DT	*Determiner*
4.	EX	*Existential there*
5.	FW	*Foreign word*
6.	IN	*Preposition or subordinating conjunction*
7.	JJ	*Adjective*
8.	JJR	*Adjective, comparative*
9.	JJS	*Adjective, superlative*
10.	LS	*List item marker*
11.	MD	*Modal*
12.	NN	*Noun, singular or mass*
13.	NNS	*Noun, plural*
14.	NNP	*Proper noun, singular*
15.	NNPS	*Proper noun, plural*
16.	PDT	*Predeterminer*
17.	POS	*Possessive ending*
18.	PRP	*Personal pronoun*
19.	PRP$	*Possessive pronoun*
20.	RB	*Adverb*
21.	RBR	*Adverb, comparative*
22.	RBS	*Adverb, superlative*
23.	RP	*Particle*
24.	SYM	*Symbol*
25.	TO	*to*
26.	UH	*Interjection*
27.	VB	*Verb, base form*
28.	VBD	*Verb, past tense*
29.	VBG	*Verb, gerund or present participle*
30.	VBN	*Verb, past participle*

No.	*Tag*	*Deskripsi*
31.	VBP	*Verb, non-3rd person singular present*
32.	VBZ	*Verb, 3rd person singular present*
33.	WDT	*Wh-determiner*
34.	WP	*Wh-pronoun*
35.	WP$	*Possessive wh-pronoun*
36.	WRB	*Wh-adverb*

Pada tugas akhir ini, *POS Tagging* digunakan untuk mengindentifikasi *POS* yang ada pada data tidak terstruktur untuk menmbantu menemukan makna pada kata tersebut. Contoh hasil dari *POS Tagging* dapat dilihat pada



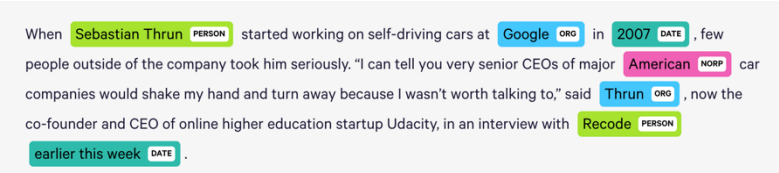
Gambar 2.1 Contoh hasil *POS Tagging*

2.5 *NER (Named Entity Recognition)*

Named Entity Recognition (NER) atau *Name Entity Recognition and Classification (NERC)* adalah salah satu komponen utama dari information extration yang bertujuan untuk mendeteksi dan mengklasifikasikan namedentity pada suatu teks.

NER umumnya digunakan untuk mendeteksi nama orang, nama tempat dan organisasi dari sebuah dokumen, tetapi dapat juga diperluas untuk identifikasi gen, protein dan lainnya sesuai kebutuhan [6].

NER bermanfaat dalam banyak aplikasi NLP (*Natural Language Processing*) seperti *question-answering*, rangkuman dan sistem dialog. NER juga berkaitan *task information extraction* lainnya seperti dengan relation detection, event detection dan temporal analysis. NER dapat diselesaikan dengan pelabelan urutan kata statistik (*statisctical sequence-labeling*) yang mendeteksi batas atau segmen dan tipe dari named-entity. Fitur yang dapat digunakan untuk learning antara lain: *shape* (*uppercase* atau *lowercase*, penggunaan angka), kata dikiri dan dikanan, jenis kata, apakah kata ada di dalam kamus atau *gazetter, predictive words dan N-Gram*. Setelah fitur dikumpukan, pelabelan dapat diselesaikan dengan menggunakan *Hidden Markov Model* atau *Maximum Entropy Model* [6]. *Named Entity Recognition (NER)* pada tugas akhir ini digunakan untuk mengekstraksi entitas-entitas



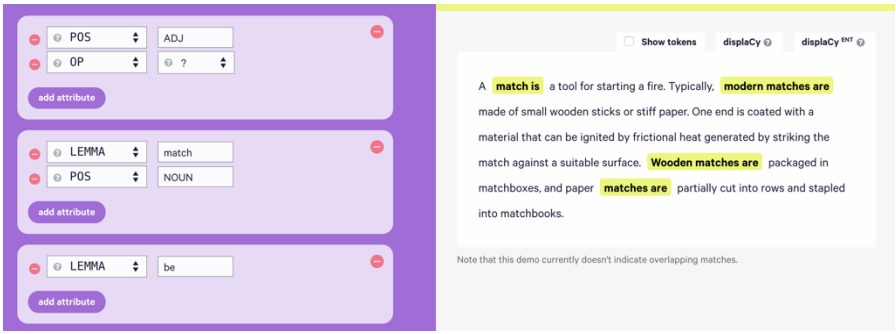
Gambar 2.2 Contoh hasil *NER*

yang akan digunakan untuk menentukan tujuan wisata. Contoh dari hasil *NER* dapat dilihat di Gambar 2.2.

2.6 Rule Based Matching

Rule Based Matching adalah mencocokkan token, frasa dan entitas kata dan kalimat sesuai dengan beberapa pola yang telah ditetapkan bersama dengan fitur seperti bagian-of-speech, jenis entitas, parsing dependensi, lemmatization dan banyak lagi. Tidak

hanya ini, tetapi ini juga mendukung pola ekspresi reguler yang membuat pendekatan pencocokan berbasis aturan spaCy memiliki



Gambar 2.3 Contoh hasil *Rule Based Matching*

keunggulan di antara perpustakaan perangkat lunak NLP lain yang tersedia saat ini [7]. *Rule Based Matching* pada tugas akhir ini digunakan untuk mengekstraksi informasi dari hasil *POS* seperti mengindentifikasi objek atau lokasi. Contoh hasil penggunaan *Rule Based Matching* dapat dilihat pada

2.7 spaCy

spaCy adalah sebuah *library* untuk pemrosesan bahasa alami tingkat lanjut, yang ditulis dalam bahasa pemrograman Python dan Cython. *Library* ini diterbitkan di bawah lisensi MIT dan saat ini menawarkan model statistik untuk Inggris, Jerman, Spanyol, Portugis, Prancis, Italia, Belanda dan multi-bahasa NER, serta tokenization untuk berbagai bahasa lainnya[8].

spaCy berfokus pada penyediaan perangkat lunak untuk pr oduksi.Pada versi 1.0, spaCy juga mendukung alur kerja pembelaj aran mendalam yang memungkinkan menghubungkan model stati stik yang dilatih oleh perpustakaan pembelajaran mesin populer s eperti TensorFlow, Keras, Scikit-learn atau PyTorch [8]. spaCy pada tugas akhir ini digunakan untuk pembuatan model *POS Tagging* dan *Named Entity Recognition (NER)* dan untuk contoh

penggunaan dari *library* spaCy ini dapat dilihat pada yang menjelaskan penggunaan model *POS Tag* yang dibuat dengan *library* spaCy untuk melakukan *POS Tagging* terhadap suatu kalimat.

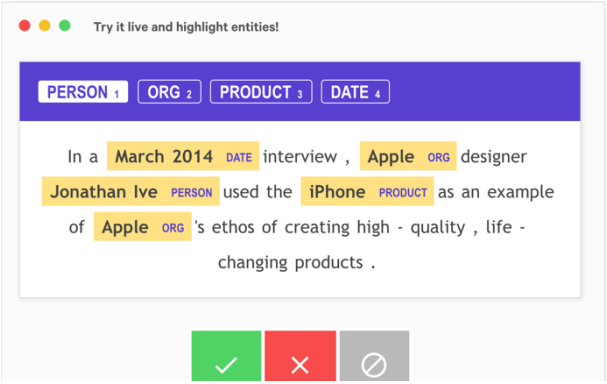
Gambar 2.4 Contoh spaCy

2.8 Prodigy

Prodigy adalah sebuah alat bantu anotasi yang digunakan untuk melakukan anotasi sendiri. Prodigy dapat mengenali entitas, mendeteksi maksud atau mengklasifikasi gambar. Prodigy dapat membantu untuk melatih dan mengevaluasi model milik pribadi sehingga dapat lebih cepat. Pengguna dapat memperbarui model secara realtime untuk membangun sistem yang lebih kompleks [8].

Aplikasi web Prodigy memungkinkan pengguna membuat anotasi teks, entitas, klasifikasi, dan gambar langsung dari broser pengguna atau bahkan di perangkat seluler pengguna. UI modernnya membuat pengguna untuk tetap fokus. Prodigy dapat memperbarui model pengguna secara *realtime* dan dapat memilih pertanyaan paling penting untuk diajukan selanjutnya [8].

Prodigy pada tugas akhir ini digunakan untuk membantu menganotasi data teks training yang digunakan untuk pembuatan model *POS Tagging* dan *NER.* Contoh dari penggunaan Prodigy adalah untuk menganotasi data dengan label- label tertentu seperti dapat dilihat pada Gambar 2.5



Gambar 2.5 Contoh penggunaan *Prodigy*

2.9 Flask

Flask adalah *kerangka kerja aplikasi web mikro* yang ditulis dalam bahasa pemrograman Python dan berdasarkan *Werkzeug toolkit* dan *template engine Jinja2* [9].

Flask disebut *micro framework* karena tidak membutuhkan alat-alat tertentu atau pustaka. Flask tidak memiliki *database abstraction layer*, *validasi form*, atau komponen lain di mana sudah ada pustaka pihak ketiga yang menyediakan fungsi umum. Namun, Flask mendukung ekstensi yang dapat menambahkan fitur aplikasi seolah-olah mereka diimplementasikan dalam Flask itu sendiri. Ekstensi yang ada untuk *object-relational mapper*, *validasi form*, penanganan unggahan, berbagai teknologi otentikasi terbuka, dan beberapa alat-lata yang terkait kerangka umum. Ekstensi diperbarui jauh lebih teratur daripada inti program Flask [9]. Pada tugas akhir ini, Flask digunakan untuk menghubungkan model- model yang sudah dibuat untuk ditampilkan dalam aplikasi web.

2.10 K-Means

K-Means Clustering merupakan salah satu metode data clustering non hirarki yang mengelompokan data dalam bentuk satu atau lebih *cluster*. Data yang memiliki karakteristik yang sama dikelompokan dalam satu *cluster* dan data yang memiliki karakteristik yang berbeda dikelompokan dengan *cluster* yang lain sehingga data yang berada dalam satu *cluster* memiliki tingkat variasi yang kecil [10].

Metode *K-Means* digunakan pada tugas akhir ini dikarenakan metode *cluster* ini cocok untuk data dengan ukuran besar karena memiliki kecepatan yang lebih tinggi dibandingkan metode hierarki yang mana data teks yang digunakan pada tugas akhir ini merupakan data dengan jumlah yang besar. *K-Means* adalah salah satu algoritma terkenal dalam clustering, awalnya dikenal sebagai metode Forgy's dan telah digunakan secara luas di berbagai bidang termasuk *Data Mining*, analisis statistik data dan aplikasi bisnis lainnya. Untuk *K-means*, K menunjukkan jumlah *cluster*. Nilai K ditentukan oleh pemakai atau *user*. Untuk kasus dimana ada pertimbangan dari ahli yang kompeten atau *expert* di bidangnya, nilai K akan mudah di tentukan. Tetapi sering sekali terjadi bahwa nilai K ini harus ditentukan dengan melihat pada data (tanpa ada pertimbangan dari expert) [10]. Algoritma untuk melakukan K-Means adalah sebagai berikut :

1. Pilih K buah titik *centroid* secara acak
2. Kelompokkan data sehingga terbentuk K buah cluster dengan titik centroid yang telah dipilih sebelumnya
3. Perbaharui nilai titik centroid
4. Ulangi langkah 2 dan 3 sampai nilai dari titik *centroid* tidak lagi berubah

Pada tugas akhir ini, metode *K-Means* digunakan untuk meng-*cluster* entitas-entitas untuk menentukan tujuan wisata di Indonesia berdasarkan kategori dari tujuan wisata.

2.11 K-Nearest Neighbor

K-nearest neighbor adalah salah satu contoh *instance-based learning*, dengan set data pelatihan (training set) disimpan, sehingga klasifikasi untuk record baru yang belum terklasifikasi dapat ditemukan hanya dengan membandingkannya dengan record paling mirip dalam training set (Larose 2005).Tahap pelatihan algoritma ini hanya menyimpan vektor fitur dan label kelas dari sampel pelatihan (MIR11). Algoritma *KNearest Neighbor* bersifat sederhana, bekerja dengan berdasarkan kemiripan dari sampel uji (*testing sample*) ke sampel latih (*training sample*) untuk menentukan *K -Nearest Neighbor* nya. Setelah mengumpulkan *K Nearest Neighbor*, kemudian diambil mayoritas dari *K-Nearest Neighbor (KNN)* untuk dijadikan prediksi dari sample uji. *KNN* memiliki beberapa kelebihan yaitu tangguh terhadap training data yang *noise* dan efektif apabila data latih nya besar. Pada fase training, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data training sample. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk testing data atau yang klasifikasinya tidak diketahui. Jarak dari vektor baru yang ini terhadap seluruh vektor training sample dihitung dan sejumlah K buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik- titik tersebut [10].

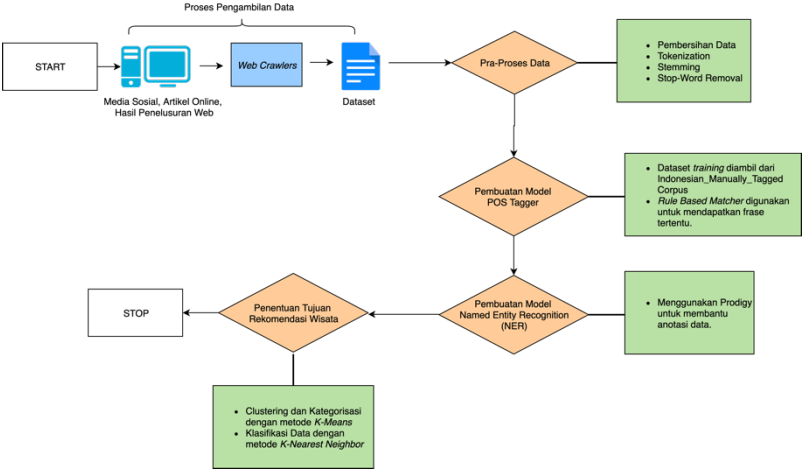
Prinsip kerja *K-Nearest Neighbor (KNN)* adalah mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga (*neighbor*) terdekatnya dalam data pelatihan [10]. Pada tugas akhir ini, klasifikasi dengan *K-Nearest Neighbor (KNN)* digunakan untuk mengklasifikasikan tujuan wisata Indonesia berdasarkan lokasi sesuai dengan provinsi. Adapun metode ini digunakan pada tugas akhir ini dikarenakan *KNN* dapat menangani data-data yang hilang dari hasil *NER* pada dataset.

BAB III
ANALISIS DAN PERANCANGAN

Pada bab ini akan dijelaskan mengenai analisis dan perancangan sistem tugas akhir yang meliputi tahap perancangan data dan perancangan proses pembuatan sistem.

3.1 Analisis Metode Secara Umum

Pada tugas akhir ini akan dibangun suatu sistem yang dap at memberikan rekomendasi pencarian kepada para wisatawan di Indonesia yang hendak mencari tujuan wisata berdasarkan data tidak terstruktur berupa artikel atau berita online. Proses-proses yang dilakukan dalam pengimplementasian sistem ini meliputi pengambilan data teks wisata, pra-proses teks, pembuatan model *Part-of-Speech (POS) Tagger* Bahasa Indonesia, pembuatan model *NER(Named Entity Recognition)* Bahasa Indonesia, serta pembuatan model clustering dan klasifikasi menggunakan K-Means dan KNN (K-Nearest Neighbor). Diagram alur proses metode dapat dilihat pada Gambar 3.1



Gambar 3.1 Diagram Alir Metode Secara Umum

Semua dataset yang digunakan berasal dari data teks online (artikel, review, berita) kecuali dataset Indonesian_Manually_Tagged_Corpus [11]. Tahapan pertama adalah melakukan pra-proses pada dataset meliputi *stopword removal, case folding* dan menghapus *whitespace*. *Stopword removal* dan *case folding* dilakukan ketika kalimat akan dilakukan POS Tagging. Sedangkan menghapus whitespace dilakukan ketika kalimat akan menggunakan model POS Tagger atau model NER bertujuan agar kalimat dapat mendapatkan *entity* maupun POS yang sesuai.

Pembuatan model POS Tagger Bahasa Indonesia dilakukan dengan spaCy. Dataset yang berasal dari Indonesian_Manually_Tagged_Corpus [11] berupa kalimat-kalimat yang disertai POS sesuai dengan format Pen Tree Bank di tiap katanya. Format pada Indonesian_Manually_Tagged_Corpus diubah ke format data spaCy agar dapat digunakan sebagai data training untuk model yang dibuat. Model yang sudah dibuat digunakan sebagai model POS Tagger Bahasa Indonesia yang siap digunakan.

Tahapan selanjutnya adalah membangun model *NER (Named Entity Recognition)*. Pembangunan model *NER* dengan menggunakan alat bantu anotasi Prodigy dan spaCy. Dikarenakan belum ada model *NER* Bahasa Indonesia dengan spAcy, pada Tugas Akhir ini penulis menggunakan *language ID* yang sudah disediakan oleh spaCy untuk melakukan anotasi secara manual menggunakan Prodigy. Hasil anotasi manual di ekspor untuk digunakan pada pembangunan model *NER* awal. Model *NER* awal ini yang selanjutnya dilatih dengan cara menambahkan anotasi dataset baru menggunakan Prodigy.

Tahapan pembuatan model clustering dan klasifikasi digunakan dengan metode K-Means dan KNN (K-Nearest Neighbor) yang akan mengelompokkan data teks menjadi suatu kategori wisata dan lokasi sesuai provinsi.

3.2 Perancangan Data

Pada subbab ini akan menjelaskan proses perancangan data. Data yang digunakan adalah data teks online berupa artikel, berita, maupun review tentang tujuan wisata di Indonesia. Untuk pengambilan data tersebut digunakan aplikasi *web crawler* berupa *Octoparse*.

Jumlah *dataset* yang digunakan adalah 1.200 kalimat yang dibagi menjadi 600 dataset untuk proses *training* dan sisanya digunakan untuk proses *testing*. Contoh *dataset* dapat dilihat pada *Tabel 3.1*.

Tabel 3.1 Dataset Hasil Crawling

Teks
Gunung Bromo adalah ikon dari Jawa Timur yang sudah terkenal hingga ke mancanegara.
Setiap tahun, ribuan turis asing dari kawasanÔøΩEropaÔøΩberdatangan ke tempat wisata di Jawa Timur ini, terutama saat musim panas. Mereka ingin melihat sendiri seperti apa kemagisan dariÔøΩGunung BromoÔøΩyang memiliki ketinggian 2.329 mdpl ini.
Gunung yang terletak di Probolinggo ini menawarkan 3 pesona yang tidak bisa ditolak dengan mudah.
Pertama adalah lautan pasir yang luas.
Berjalan dari parkiran hingga ke puncak gunung akan membuat Anda seperti berada di gurun pasir.

3.3 Perancangan Proses

Pada subbab ini akan dijelaskan mengenai peracangan proses yang dilakukan untuk setiap tahap pembuatan tugas akhir ini.

3.3.1 *POS Tagging*

POS Tagging digunakan untuk mendapatkan kata-kata dengan *POS* tertentu. Proses ini digunakan untuk mendapatkan kata-kata sesuai dengan *POS* atau sifat katanya. *POS* yang digunakan dapat dilihat pada Tabel 3.2.

Tabel 3.2 POS Tag

POS	Deskripsi	Contoh
ADJ	adjective	besar, tua, menarik, asik
ADP	adposition	di, ke, sedang
ADV	adverb	sangat, besok, bawah, di mana, sana
CONJ	conjunction	dan, atau, tapi
DET	determiner	suatu
NOUN	noun	udara, pohon, air
NUM	numeral	1, 2017, satu, dua
PART	particle	tidak
PRON	pronoun	Saya, kamu, dia
PROPN	proper noun	Banyuwangi, Jakarta, Budi, Syifa
PUNCT	punctuation	., (,), ?
SCONJ	subordinating conjunction	sementara, sesudah, karena, bahkan
SYM	symbol	$, %, §, ©, +, −, ×, ÷, =, :), 😝

POS	Deskripsi	Contoh
VERB	verb	berlari, mengerjakan, memakan, melakukan
X	other	sfpksdpsxmsa

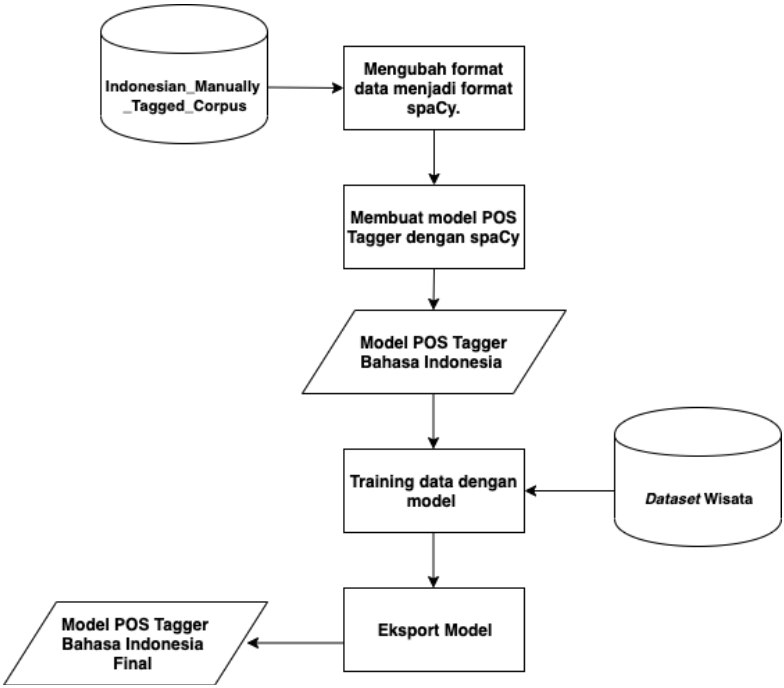
Pembuatan model POS Tagging dengan menggunakan data training POS Tag Bahasa Indonesia milik Fam Rashel [11]. yang ditambah dengan 600 *dataset* dari data teks online. Proses pembuatan model POS Tagging dapat dilihat pada Gambar 3.2 dan contoh hasil dari POS Tagging dapat dilihat pada Tabel 3.3.

Tabel 3.3 Contoh hasil POS Tag

Input	Output POS Tag
Kawah Ijen telah berhasil menjadi ikon kedua dari Jawa Timur yang ketenarannya tidak kalah dengan Bali. Terletak di kawasan Banyuwangi, kawah dari gunung yang terus mengeluarkan belerang ini menjadi tujuan wisatawan yang ingin merasakan pendakian kecil sembari menyaksikan bentang alam yang menakjubkan.	Kawah NOUN Ijen PROPN telah AUX berhasil VERB menjadi VERB ikon NOUN kedua NUM dari ADP Jawa PROPN Timur PROPN yang SCONJ ketenarannya ADV tidak X kalah NOUN dengan ADP Bali PROPN . PUNCT Terletak VERB di ADP kawasan NOUN

Input	Output POS Tag
	Banyuwangi PROPN , PUNCT kawah NOUN dari ADP gunung NOUN yang SCONJ terus AUX mengeluarkan VERB belerang NOUN ini PRON menjadi VERB tujuan NOUN wisatawan NOUN yang SCONJ ingin ADV merasakan VERB pendakian NOUN kecil ADJ sembari ADV menyaksikan VERB bentang VERB alam NOUN yang SCONJ menakjubkan VERB . PUNCT

Proses pembuatan model *POS Tagging* diawali dengan mengubah format pada dataset yang digunakan yaitu Indonesian_Manually_Tagged_Corpus menjadi format yang sesuai dengan spaCy sebagai library untuk membuat model POS tag, setelah dataset sudah diubah formatnya sesuai dengan format



Gambar 3.2 Diagram Alir POS Tag

spaCy dibuat sebuah model *POS Tag* dengan menggunakan tag yang sesuai dengan yang tertera pada Tabel 3.2. Setelah model POS Tag Bahasa Indonesia sudah berhasil dibuat, selanjutnya adalah proses *training* data dengan dataset teks wisata yang sudah dianotasi agar model POS Tag dapat mengenali lebih banyak kata-kata terutama data yang berhubungan dengan pariwisata. Tahap selanjutnya adalah mengekspor model yang sudah siap digunakan agar dapat dipanggil oleh *library* spaCy untuk melakukan *POS Tagging* terhadap dataset yang ingin diolah.

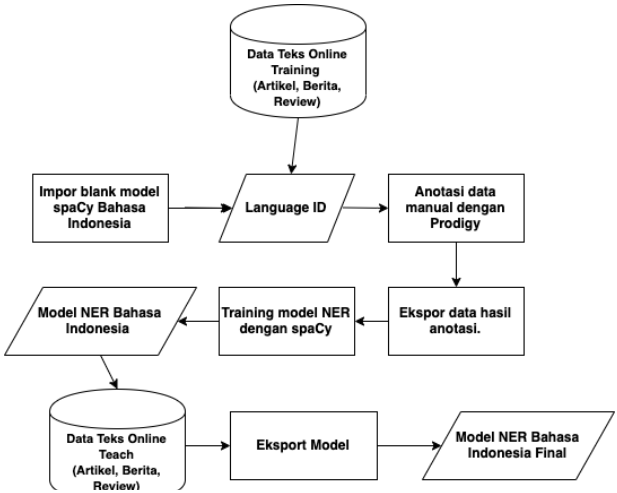
3.3.2 Named Entity Recognition (NER)

Pada tugas akhir ini , penulis menggunakan NER untuk mendeteksi kata dan frasa yang ingin ditarik untuk menjadi sebuah *entity* tertentu. Pembangunan model menggunakan *library spaCy* dan alkat bantu anotasi Prodigy. Label yang digunakan dalam pembangunan NER Bahasa Indonesia dapat dilihat pada Tabel 3.4

Tabel 3.4 Label Entitas NER

Entity	Deskripsi
LOKASI	Lokasi dari tujuan wisata seperti kota, kabupaten.
OBJEK	Objek wisata, nama tempat wisata
KETERANGAN	Frasa opini dari tempat wisata seperti sejuk, indah, adem, dll.
KATEGORI	Kata-kata untuk menmgkategorikan tempat wisata seperti pantai, gunung, laut, dll.

Untuk membuat model NER Bahasa Indonesia baru, dataset Bahasa Indonesia dianotasi manual menggunakan Prodigy dengan memanggil *blank model Id* dari spaCy. Kemudian hasil anotasinya di ekspor untuk dijadikan data *pre-trained* pada pembuatan model. Setelah model *pre-trained* jadi, model tersebut akan dilatih dengan untuk lebih mengenali kata kata yang lainnya dengan menggunakan Prodigy. Lalu akan dieskpor sehingga menjadi model *NER* yang siap digunakan. Proses pembuatan model *NER* Bahasa Indonesia dapat dilihat pada Gambar 3.3 dan contoh hasil *NER* dapat dilihat pada Tabel 3.5.



Gambar 3.3 Diagram Alur Named Entity Recognition (NER)

Tabel 3.5 Contoh hasil *NER*

Input	Output NER
Kawah Ijen telah berhasil menjadi ikon kedua dari Jawa Timur yang ketenarannya tidak kalah dengan Bali. Terletak di kawasan Banyuwangi, kawah dari gunung yang terus mengeluarkan belerang ini menjadi tujuan wisatawan yang ingin merasakan pendakian kecil sembari menyaksikan bentang alam yang menakjubkan.	[(kawah, 'objek'), (ijen, 'objek'), (jawa, 'lokasi'), (timur, 'lokasi'), (bali, 'lokasi'), (., 'lokasi'), (banyuwangi, 'lokasi'), (menakjubkan, 'keterangan')]

3.3.3 Rule Based Matching

Proses *rule-based matching* digunakan untuk mendapatkan frasa tertentu (sesuai rule) untuk digunakan pada

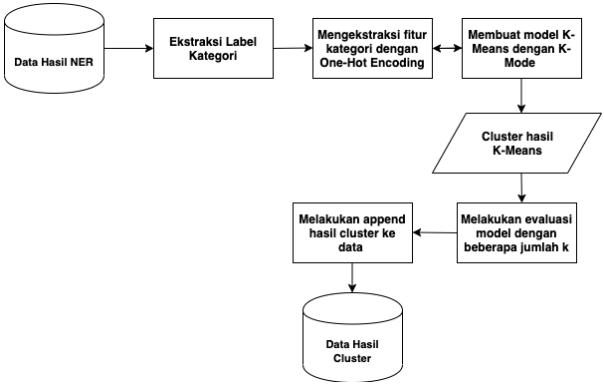
proses *POS Tagging*. Pada tugas akhir ini, penulis menggunakan *library* spaCy untuk melakukan *rule-based matching* untuk menangkap objek. *Rule* didapatkan dari hasil proses *POS Tagging* dimana objek wisata terdeteksi sebagai PROPN, sehingga rule yang dibuat untuk membantu menangkap objek wisata adalah dengan menyatukan *POS* yang terdeteksi sebagai PROPN. Daftar *rule* yang digunakan pada tahap ini dapat dilihat pada Tabel 3.6.

Tabel 3.6 Daftar *Rule*

Rule	Fungsi
PROPN + PROPN	Mendeteksi objek wisata

3.3.4 *Clustering*

Selanjutnya dilakukan *clustering* dari hasil *Named Entity Recognition (NER)* dengan label kategori untuk menentukan



Gambar 3.4 Diagram Alir Proses *Clustering*

kategori dari setiap objek wisata. Metode *clustering* yang digunakan pada tugas akhir ini adalah *K-Means Clustering*. *K-Means* yang digunakan adalah metode *K-Mode* dimana merupakan metode clustering untuk data yang bukan nominal, melainkan data

kategorik. Proses *clustering* diawali dengan data hasil proses *Named Entity Recognition (NER)* diekstraksi untuk label kategori untuk selanjutnya dilakukan proses seleksi fitur dengan metode *One-Hot Encoding*, setelah melakukan proses seleksi fitur dibuatlah model *K-Means Clustering* dengan bantuan *library* K-Mode yang menghasilkan hasil *cluster-cluster* yang digunakan sebagai kategori untuk wisata. Tahap selanjutnya adalah tahap eveluasi terhadap model *K-Means* yang sudah dibuat lalu melakukan penyimpanan terhadap data hasil cluster untuk tahap selanjutnya.

Proses *clustering* dapat dilihat pada Gambar 3.4 dan hasil dari proses ini dapat juga dilihat pada Tabel 3.7

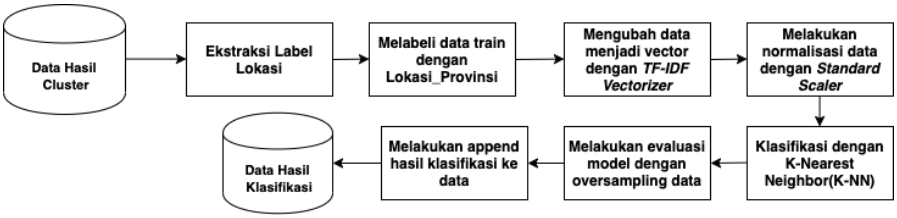
Tabel 3.7 Hasil Proses Clustering

Hasil *NER*	Kategori	*Clusters*
[(gunung, 'objek'), (bromo, 'objek'), (gunung, 'objek'), (bromo, 'objek'), (jawa, 'lokasi'), (timur, 'lokasi'), (terkenal, 'keterangan'), (jawa, 'lokasi'), (timur, 'lokasi'), (kemagisan, 'keterangan'), (gunung, 'objek'), (bromo, 'objek'), (gunung, 'kategori'), (probolinggo, 'lokasi'), (pesona, 'keterangan')]	gunung	1
[(kawah, 'objek'), (ijen, 'objek'), (jawa, 'lokasi'), (timur, 'lokasi'), (bali, 'lokasi'), (., 'lokasi'), (banyuwangi, 'lokasi'), (menakjubkan, 'keterangan')]	gunung	1

3.3.5 Klasifikasi

Proses klasifikasi dilakukan setelah data sudah dicluster sesuai dengan kategori wisata. Klasifikasi pada tugas akhir ini dilakukan untuk mengklasifikasikan data berdasarkan lokasi secara provinsi agar membantu *searching* berdasarkan provinsi tempat wisata. Metode klasifikasi yang digunakan pada tugas akhir ini

adalah metode *K-Nearest Neighbor (K-NN)* dengan metode pemilihan fitur dengan metode *TF-IDF-Vectorizer* untuk membantu mengubah data teks menjadi vector. Proses klasifikasi diawali dengan data hasil proses *K-Means Clustering* pada tahap sebelumnya di ekstraksi untuk label lokasi yang akan digunakan sebagai fitur untuk proses ini, selanjutnya data yang sudah di ekstraksi dilabeli secara manual untuk fitur provinsi dari lokasi yang sudah di ekstraksi. Selanjutnya adalah untuk mengubah data menjadi vector dengan metode *TF-IDF Vectorizer* untuk membantu proses seleksi fitur, data yang sudah diubah akan dilakukan proses normalisasi data dengan *Standard Scaler* . Seteleh proses seleksi fitur sudah selesai, dibuatlah model klasifikasi dengan *K-Nearest Neighbor.* Tahap selanjutnya adalah untuk melakukan evaluasi model yang sudah dibuat dan melakukan penyimpanan data hasil klasifikasi. Proses klasifikasi dapat dilihat pada Gambar 3.5 dan hasil dari proses ini dapat dilihat pada



Gambar 3.5 Diagram Alur Klasifikasi dengan *K-Nearest Neighbor (K-NN)*

Tabel 3.8 Hasil Proses Klasifikasi

Hasil NER	Lokasi	Lokasi Provinsi	Klasifikasi
[(gunung, 'objek'), (bromo, 'objek'), (gunung, 'objek'), (bromo, 'objek'), (jawa, 'lokasi'), (timur, 'lokasi'),	probolinggo	jawa timur	1

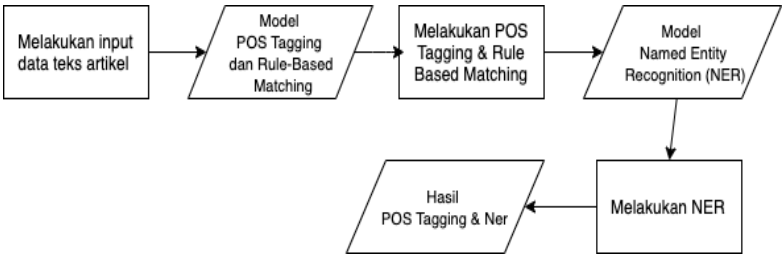
Hasil NER	Lokasi	Lokasi Provinsi	Klasifikasi
(terkenal, 'keterangan'), (kemagisan, 'keterangan'), (gunung, 'objek'), (bromo, 'objek'), (gunung, 'kategori'), (probolinggo, 'lokasi'), (pesona, 'keterangan')]			

3.3.6 Evaluasi dan Uji Coba

Pada tahap evaluasi dilakukan uji coba untuk model *POS Tagging* dan *Named Entity Recognition (NER)*, hasil prediksi model *clustering* dengan membandingkan *silhouette score* dengan beberapa nilai k. Untuk hasil klasifikasi, evaluasi hasil prediksi model dibandingkan dengan *groundtruth.* Agar mempermudah, dibuat sebuah confusion matrix untuk mendapatkan nilai akurasi, *recall, precision* dan *F-Measure* dari setiap evaluasi model.

3.3.7 Visualisasi

Visualisasi pada tugas akhir ini dilakukan dengan menampilkan demo simulasi aplikasi metode *POS Tagging* dan *Named Entity Recognition (NER)* dengan melakukan input artikel teks berita dan menghasilkan data yang terstruktur. Visualisasi selanjutnya pada tugas akhir ini adalah untuk menampilkan query hasil *clustering* dan klasifikasi. Database yang digunakan untuk menyimpan hasil adalah dengan menggunakan MongoDB. Pada menunjukkan diagram alir untuk simulasi aplikasi metode *POS Tagging* dan *Named Entity Recognition (NER)*.



Gambar 3.6 Diagram Alir Visualisasi metode *POS Tag & NER*

Sistem uji aplikasi untuk menampilkan hasil visualisasi dimulai dengan melakukan input data teks artikel objek wisata yang selanjutnya akan di proses dengan model *POS Tagging* dan *Named Entity Recognition (NER)* yang akan ditampilkan hasilnya dengan bantuan *library* displaCy. Pada halaman query, selanjutnya dapat dilakukan query untuk hasil *clustering* dan klasifikasi sesuai dengan kategori wisata dan lokasi dari objek wisata.

BAB IV
IMPLEMENTASI SISTEM

Bab ini membahas implementasi dari perancangan sistem sesuai dengan perancangan yang telah dibuat. Bab ini juga akan merinci tools yang digunakan pada tugas akhir.

4.1 Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan tugas akhir ini memiliki spesifikasi perangkat keras dan perangkat lunak yang ditunjukkan oleh Tabel 4.1.

Tabel 4.1 Spesifikasi Perangkat

Perangkat	Spesifikasi
Perangkat Keras	• Laptop : Macbook Pro 2018 15-inch • Prosesor: Intel® Core™ i7-7700U CPU @ 2.20GHz (6CPUs), ~3.6GHz • Memori: 256 GB • VGA: Radeon Pro 555x memori GDDR5 4 GB
Perangkat Lunak	• Sistem Operasi macOs Mojave 64-bit • Perangkat Pengembang Visual Studio Code, Sublime Text, Microsoft Word, Jupyter Notebook, Google Chrome, MongoDB Community

Tabel 4.2 Tools

No	Tools	Deskripsi
1	Python	Bahasa *Python* digunakan untuk menangani *task Natural Language Processing* (NLP).
2	Prodigy	*Tools* Anotasi untuk melabeli data untuk pembuatan model *POS Tagging* dan *NER.*
3	spaCy	*Library Python* yang digunakan untuk membuat model *POS Tagging* dan *Named Entity Recognition (NER).*
4	Scikit-Learn	*Library* yang digunakan untuk melakukan klasifikasi.
5	K-Mode	*Library* yang digunakan untuk melakukan *clustering.*
6	Flask	*Library* yang digunakan untuk menampilkan python ke dalam aplikasi web.

4.2 Implementasi Proses

Implementasi proses merupakan tahap implementasi pada perancangan proses yang sebelumnya sudah dijelaskan pada bab analisis dan perancangan. Implementasi proses meliputi pembuatan model *POS Tag,* model *Named Entity Recognition (NER)*, proses clustering dengan *K-Means* dan proses klasifikasi dengan *K-Nearest Neighbor* serta implementasi visualisasi dan penyimpanan data.

4.2.1 Implementasi Pembuatan Model *POS Tagger* Bahasa Indonesia

Model *POS Tagger* Bahasa Indonesia dibangun dengan menggunakan library spaCy. Proses pembuatan model *POS Tagger* diawali dengan mengubah format *tag* dari format yang ada pada *PenTree Bank* ke format *tag* POS pada umunnya, selanjutnya dilakukan input data training yang sudah diubah ke format `.pkl`. Data yang di input merupakan data training yang berasal dari Indonesian_Manually_Tagged_Corpus [11]. Membuat model *POS Tag* dengan membangun pipeline dan train tagger dengan menginput bahasa dan pipeline tagger, selanjutnya dilakukan batch up contoh dengan menggunakan spaCy *minibatch*, setelah proses ini dilakukan tes training terhadap model yang sudah dibuat dengan input data teks baru, lalu dilakukan penyimpanan model yang sudah dibuat. Kode sumber pembuatan model POS Tagger dapat dilihat pada Kode Sumber 4.1 dan hasil dari proses POS Tagging dengan model yang sudah dibuat dapat dilihat pada Tabel 4.3

```
1.  from __future__ import unicode_literals, print_fu
    nction
2.  import plac
3.  import random
4.  from pathlib import Path
5.  import spacy
6.  from spacy.util import minibatch, compounding
7.  import pickle
8.  #Convert tag from PenTree Bank format to common P
    OS format
9.
10. TAG_MAP = {
11.
12.             "CC": {"pos": "CONJ"},
```

```
13.
14.             "CD": {"pos": "NUM"},
15.
16.             "DT": {"pos": "DET"},
17.
18.             "FW": {"pos": "X"},
19.
20.             "IN": {"pos": "ADP"},
21.
22.             "JJ": {"pos": "ADJ"},
23.
24.             "MD": {"pos": "AUX"},
25.
26.             "NEG": {"pos": "X"},
27.
28.             "NN": {"pos": "NOUN"},
29.
30.             "NND": {"pos": "NOUN"},
31.
32.             "NNP": {"pos": "PROPN"},
33.
34.             "OD": {"pos": "NUM"},
35.
36.             "PR": {"pos": "PRON"},
37.
38.             "PRP": {"pos": "PRON"},
39.
40.             "RB": {"pos": "ADV"},
41.
42.             "RP": {"pos": "PART"},
43.
44.             "SC": {"pos": "SCONJ"},
45.
46.             "SYM": {"pos": "SYM"},
47.
48.             "UH": {"pos": "INTJ"},
49.
50.             "VB": {"pos": "VERB"},
51.
52.             "WH": {"pos": "X"},
53.
```

```
                "XX": {"pos": "X"},

                "Z": {"pos": "PUNCT"}

            }
TRAIN_DATA = []

cv = open("indonesian-manually-
tagged.pkl","rb")
TRAIN_DATA = pickle.load(cv)

print(TRAIN_DATA)

print("Let's train %d strings!"%(len(TRAIN_DATA))
)



@plac.annotations(

    lang=("ISO Code of language to use", "option"
, "l", str),

    output_dir=("Optional output directory", "opt
ion", "o", Path),

    n_iter=("Number of training iterations", "opt
ion", "n", int),

)

def main(lang="id", output_dir=('/Users/daniseyy/
pos-tag'), n_iter=25):

#Membuat model, membangun pipeline dan train tagg
er. Training tagger baru dan membuat bahasa baru


    nlp = spacy.blank(lang)

# Menambahkan tagger ke pipeline

```

```
88.     tagger = nlp.create_pipe("tagger")
89.
90. # Menambahkan tag
91.
92.     for tag, values in TAG_MAP.items():
93.
94.         tagger.add_label(tag, values)
95.
96.     nlp.add_pipe(tagger)
97.
98.
99.
100.            optimizer = nlp.begin_training()
101.
102.            for i in range(n_iter):
103.
104.                random.shuffle(TRAIN_DATA)
105.
106.                losses = {}
107.
108.        #Melakukan batch up contoh dengan mengguna
    kan spaCy's minibatch
109.
110.                batches = minibatch(TRAIN_DATA, si
    ze=compounding(4.0, 32.0, 1.001))
111.
112.                for batch in batches:
113.
114.                    texts, annotations = zip(*batc
    h)
115.
116.                    nlp.update(texts, annotations,
     sgd=optimizer, losses=losses)
117.
118.                print("Losses", losses)
119.
120.
121.
122.        # Tes trained model
123.
```

```
124.            test_text = "Gunung Bromo adalah ikon
    dari Jawa Timur yang sudah terkenal hingga ke man
    canegara."
125.
126.            doc = nlp(test_text)
127.
128.            print("Tags", [(t.text, t.tag_, t.pos_
    ) for t in doc])
129.
130.
131.
132.        #Menyimpan model ke output directory
133.
134.            nlp.to_disk("./pos-tag")
135.
136.            print("Saved model to", output_dir)
137.
138.
139.
140.        #Tes model yang baru saja disimpan
141.
142.            print("Loading from", output_dir)
143.
144.            nlp2 = spacy.load("./pos-tag")
145.
146.            doc = nlp2(test_text)
147.
148.            print("Tags", [(t.text, t.tag_, t.pos_
    ) for t in doc])
149.        if __name__ == "__main__":
150.
151.            plac.call(main)
152.
153.        from __future__ import unicode_literals, p
    rint_function
```

Kode Sumber 4.1 Implementasi POS Tag

Tabel 4.3 Hasil dari *POS Tagging*

Input	Output POS Tag
Kawah Ijen telah berhasil menjadi ikon kedua dari Jawa Timur yang ketenarannya tidak kalah dengan Bali. Terletak di kawasan Banyuwangi, kawah dari gunung yang terus mengeluarkan belerang ini menjadi tujuan wisatawan yang ingin merasakan pendakian kecil sembari menyaksikan bentang alam yang menakjubkan.	Kawah NOUN Ijen PROPN telah AUX berhasil VERB menjadi VERB ikon NOUN kedua NUM dari ADP Jawa PROPN Timur PROPN yang SCONJ ketenarannya ADV tidak X kalah NOUN dengan ADP Bali PROPN . PUNCT Terletak VERB di ADP kawasan NOUN Banyuwangi PROPN , PUNCT kawah NOUN dari ADP gunung NOUN yang SCONJ terus AUX mengeluarkan VERB belerang NOUN ini PRON menjadi VERB

Input	Output POS Tag
	tujuan NOUN wisatawan NOUN yang SCONJ ingin ADV merasakan VERB pendakian NOUN kecil ADJ sembari ADV menyaksikan VERB bentang VERB alam NOUN yang SCONJ menakjubkan VERB . PUNCT

4.2.2 Implementasi Pembuatan Model *Named Entity Recognition (NER)*

Pada sub-bab ini akan dijelaskan proses pembuatan model NER Bahasa Indonesia. Dataset yang digunakan untuk membuat model ini berasa dari data teks online seperti artikel, berita, atau review tujuan wisata. Dataset berjumlah 1.200 data dan dibagi menjadi dua, yaitu 600 data untuk *training* dan 600 data untuk data *testing.*

4.2.2.1 Membuat *Database* di Prodigy

Database digunakan untuk menyimpan hasil anotasi di Prodigy. Cara membuat *database* dapat dilihat pada Kode Sumber 4.2 Pada Kode Sumber 4.2 database diberi nama train-ta.

```
python -m prodigy dataset train_ta “Train Data NER”
```

Kode Sumber 4.2 Pembuatan Database di Prodigy

4.2.2.2 Anotasi Manual dengan Prodigy

Anotasi manual dilakukan untuk memberi label pada setiap kata secara manual satu per satu. Label yang digunakan pada Tugas Akhir ini berjumlah 4. Kode Sumber 4.3 yang digunakan untuk menganotasi secara manual terdapat pada Kode Sumber 4.3

```
python -m prodigy ner.manual train_ta /Users/daniseyy/id-
model /Users/daniseyy/train-ta.txt –label “LOKASI,
OBJEK, KATEGORI, KETERANGAN”
```

Kode Sumber 4.3 Anotasi Manual Prodigy

Pada Kode Sumber dijelaskan bahwa database bernama train_ta. Model yang digunakan merupakan model *language ID* Bahasa Indonesia terletak di /Users/daniseyy/id-model. Data training diletakkan di /Users/daniseyy/train-ta.txt. Label yang digunakan yaitu LOKASI, OBJEK, KATEGORI, KETERANGAN.

4.2.2.3 Ekspor Hasil Anotasi

Setelah melakukan anotasi 500 dataset untuk data *training* maka hasil tersebut perlu di ekspor dalam bentuk file JSONL menyesuaikan dengan format spaCy. Kode Sumber dapat dilihat di Kode Sumber 4.4 . *Dataset* yang terekspor hanyalah dataset yang Accepted. Sehingga jumlah hasil ekspor tidak sama dengan jumlah dataset yang dianotasi manual.

Kode Sumber 4.4 untuk melakukan ekspor hasil anotasi di Prodigy menggunakan command ner.gold-to-spacy. *Database* yang ingin diekspor isinya adalah train_ta. Directory penyimpanan file hasil anotasi pada /Users/daniseyy/train-ta-fix.jsonl

```
python -m prodigy ner.gold-to-spacy train_ta
/Users/daniseyy/train-ta-fix.jsonl
```

Kode Sumber 4.4 Ekspor Hasil Anotasi Prodigy

4.2.2.4 Membuat Model NER (Named Entity Recognition)

Pembuatan model dilakukan dengan bantuan library spaCy. Model dibuat dengan melakukan training terhadap data hasil anotasi yang kemudian di export dan disimpan pada directory /Users/daniseyy/ner-wisata. Kode Sumber dapat dilihat pada Kode Sumber 4.5 . dan hasil dari model *NER* yang dibuat dapat dilihat pada Tabel 4.4

```
1.  #variabel TRAIN_DATA beirisi hasil anotasi pada
    subab 4.2.2.3.
2.  import spacy
3.  import random
4.
5.
6.  TRAIN_DATA = []
7.
8.
9.  def train_spacy(data,iterations):
10.     TRAIN_DATA = data
11.     nlp = spacy.blank('id')  # create blank Langu
    age class
12.     # create the built-
    in pipeline components and add them to the pipeli
    ne
13.     # nlp.create_pipe works for built-
    ins that are registered with spaCy
14.     if 'ner' not in nlp.pipe_names:
15.         ner = nlp.create_pipe('ner')
16.         nlp.add_pipe(ner, last=True)
17.
18.
19.     # add labels
20.     for _, annotations in TRAIN_DATA:
21.          for ent in annotations.get('entities'):

22.             ner.add_label(ent[2])
23.
```

```
    # get names of other pipes to disable them during training
    other_pipes = [pipe for pipe in nlp.pipe_names if pipe != 'ner']
    with nlp.disable_pipes(*other_pipes):  # only train NER
        optimizer = nlp.begin_training()
        for itn in range(iterations):
            print("Statring iteration " + str(itn))
            random.shuffle(TRAIN_DATA)
            losses = {}
            for text, annotations in TRAIN_DATA:

                nlp.update(
                    [text],  # batch of texts
                    [annotations],  # batch of annotations
                    drop=0.2,  # dropout - make it harder to memorise data
                    sgd=optimizer,  # callable to update weights
                    losses=losses)
            print(losses)
    return nlp


prdnlp = train_spacy(TRAIN_DATA, 474)

# Save our trained Model
modelfile = input("Masukkan nama model: ")
prdnlp.to_disk(modelfile)
  #Test your text
test_text = input("Masukkan test text: ")
doc = prdnlp(test_text)
for ent in doc.ents:
    print(ent.text, ent.start_char, ent.end_char, ent.label_)
```

Kode Sumber 4.5 Implementasi Pembuatan *NER*

Tabel 4.4 Hasil Model *NER*

Input	Output NER
Kawah Ijen telah berhasil menjadi ikon kedua dari Jawa Timur yang ketenarannya tidak kalah dengan Bali. Terletak di kawasan Banyuwangi, kawah dari gunung yang terus mengeluarkan belerang ini menjadi tujuan wisatawan yang ingin merasakan pendakian kecil sembari menyaksikan bentang alam yang menakjubkan.	[(kawah, 'objek'), (ijen, 'objek'), (jawa, 'lokasi'), (timur, 'lokasi'), (bali, 'lokasi'), (., 'lokasi'), (banyuwangi, 'lokasi'), (menakjubkan, 'keterangan')]

4.2.2.5 *Training* dengan Model NER (Named Entity Recognition)

Tahapan ini bertujuan agar model lebih banyak mengenal kalimat-kalimat sehingga dapat melakukan proses NER dengan tepat. Terdapat dua *command* untuk melakukan *training* dengan Prodigy, yaitu dengan *ner.make-gold* dan *ner.teach*. Pada *ner.make-gold,* sistem akan melakukan anotasi secara otomatis, apabila terjadi kesalahan dalam penganotasian, pengguna dapat menggantinya dengan label yang lain. Sedangkan untuk *ner.teach* pengguna hanya mengecek apakah label yang dianotasi oleh sistem sudah benar atau tidak. Pada Tugas Akhir ini penulis melakukan *ner.teach* terlebih dahulu, kemudian menggunakan ner.make-gold untuk melakukan koreksi terhadap anotasi yang tidak sesuai. Sintaks yang digunakan dapat dilihat pada Kode Sumber 4.6

```
python -m prodigy ner.teach train_ta /Users/daniseyy/ner-wisata /Users/daniseyy/train-ta.txt –loader txt –label “LOKASI, OBJEK, KATEGORI, KETERANGAN”
```

Kode Sumber 4.6 Training dengan model *NER*

Pada Kode Sumber 4.6 database yang digunakan adalah train_ta. Model yang digunakan terletak di /Users/daniseyy/ner-wisata. File yang menyimpan data training terletak di /Users/daniseyy/train-ta.txt. Serta label yang digunakan adalah LOKASI, OBJEK, KATEGORI, KETERANGAN.

4.2.2.6 Mengeskpor Model serta Melakukan Training di Prodigy

Selanjutnya dilakukan training untuk menambah anotasi-anotasi yang sesuai dengan tujuan Tugas Akhir ini. Training dilakukan dengan menggunakan salah satu fungsi Prodigy yaitu ner.batch-train. Fungsi ini akan melakukan training dan mengupdate model yang sudah ada. Kode Sumber dapat dilihat pada Kode Sumber 4.7

```
python -m prodigy ner.batch-train train-ta
/Users/daniseyy/ner-wisata /Users/daniseyy/ner-wisata –
eval-split 0.2 –n-iter 10 –batch-size 10
```

Kode Sumber 4.7 Ekspor Model di Prodigy

4.2.3 Implementasi *POS Tagging*

Implementasi *POS Tagging* digunakan untuk mendapatkan kata-kata sesuai dengan POS atau sifat katanya. Pada tugas akhir ini dilakukan *POS Tagging* dengan 17 label seperti pada Tabel Kode sumber implementasi *POS Tagging* dapat dilihat pada Kode Sumber 4.8. dan contoh hasil dari implementasi *POS Tagging* dapat dilihat pada Gambar 4.1

```
1. import spacy
```

```
from spacy import displacy
from colors import get_entity_options
#Load Model
nlp = spacy.load('/Users/daniseyy/pos-tag')
#Menerima input teks
def formartikel_post() :
text = request.form['text']
    print(text)
#Melakukan POS Tagging
    doc2 = nlp(text)
    options = get_entity_options()
#Mencetak kata dan hasil POS
    print([(X.text, X.pos_) for X in doc2])
#Memvisualisasikan hasil POS dengan DisplaCy
    svg2 = displacy.render(doc2, style='dep')
    svg2 = Markup(svg2)


    flash([svg2])
return render_template('form-artikel.html')
```

Kode Sumber 4.8 Implementasi POS Tagging

Gambar 4.1 Hasil Implementasi *POS Tag*

4.2.4 Implementasi *Rule Based Matching*

Pada pengimplementasian *rule-based matching* setelah melakukan POS Tagging. Kode sumber implementasi *rule-based matching* dapat dilihat pada Kode Sumber 4.9

```
import spacy
nlp = spacy.load('/Users/daniseyy/pos-tag')
```

```
3.   def formartikel_post():
4.   text = request.form['text']
5.      print(text)
6.   matcher = Matcher(nlp.vocab)
7.
8.   pattern = [{"POS": "PROPN", "OP": "?"}, {"POS":
    "PROPN", "OP": "?"}]
9.   matcher.add("OBJEK", None, pattern)
10.
11.  doc = nlp(text)
12.  token_match = []
13.  matches = matcher(doc)
14.  for match_id, start, end in matches:
15.      string_id = nlp.vocab.strings[match_id]
16.      span = doc[start:end]
17.      token_match.append(span.text)
18.  return token_match
```

Kode Sumber 4.9 Implementasi *Rule Based Matching*

Namun dikarenakan sudah ada *Named Entity Recognition (NER)* yang dapat menangkap entitas objek dengan model yang dibuat, maka tidak lagi dibutuhkan penggunaan *Rule Based Matching* pada bagian ini.

4.2.5 Implementasi *Named Entity Recognition (NER)*

Pengimplementasian *Named Entity Recognition (NER)* digunakan untuk mendeteksi frasa dan kata penting dari suatu entitas. Pada tugas akhir ini digunakan 4 label yang terdapat pada Tabel. Implementasi *NER* dapat dilihat pada Kode Sumber 4.10 dan untuk hasil implementasi *NER* dapat dilihat pada Gambar 4.2

```
1.  import spacy
2.  from spacy import displacy
3.  from colors import get_entity_options
```

```
#Load Model
nlp = spacy.load('/Users/daniseyy/ner-wisata')
#Menerima input teks
text = request.form['text']
    print(text)
#Melakukan deteksi NER
    doc = nlp(text)
    options = get_entity_options()
#Mencetak kata dan hasil NER
    print([(J.text, J.label_) for J in doc.ents])

#Memvisualisasikan hasil NER dengan DisplaCy
    svg2 = displacy.render(doc, style='ent')
    svg2 = Markup(svg)


    flash([svg])
return render_template('form-artikel.html')
```

Kode Sumber 4.10 Implementasi *NER*

Gambar 4.2 Hasil *NER*

4.2.5.1 Melakukan Penyimpanan hasil *NER* untuk setiap Entitas

Untuk mempermudah proses selanjutnya dilakukan penyimpanan hasil NER untuk setiap entitas. Proses ini dapat dilihat pada Kode Sumber 4.11

```
1.  import spacy
2.  from spacy import displacy
3.  from colors import get_entity_options
4.  #Load Model
5.  nlp = spacy.load('/Users/daniseyy/ner-wisata')
6.  text = request.form['text']
7.  #Membuat array kosong untuk melakukan append
8.  result= []
9.  for text in text:
10.     sentence_nlp = nlp(text)
11.     #Melakukan append hasil NER dengan kategori k
    husus
12.     #"" diisi dengan Entity yang diinginkan
13.     result.append([(word) for word in sentence_nl
    p if word.ent_type_  == ""])
14.     print([(word) for word in sentence_nlp if wor
    d.ent_type_ == ""])
```

Kode Sumber 4.11 Menyimpan hasil *NER*

4.2.6 Ekstraksi Hasil *NER*

Dikarenakan hasil *NER* yang sudah di simpan masih dalam berbentuk *array* di dalam *bracket,* maka dilakukan ekstraksi entitas dengan metode *Regular Expression.* Ekstraksi hasil *NER* dapat dilihat pada Kode Sumber 4.12

```
1.  import re
2.
3.  readstream = result
```

```
4.  stringExtract2 = re.findall(r'\(([^()]+)\)', read
    stream)
```

Kode Sumber 4.12 Ekstraksi Hasil *NER*

4.2.7 Implementasi *K-Means Clustering*

Pada sub-bab ini akan dijelaskan proses *clustering* data hasil *NER* untuk menentukan kategori wisata.

4.2.7.1 Pra-Proses Data

Pada tahap ini, dilakukan pre-processing terhadap data hasil NER yaitu proses *case folding* , *remove punctuation* , menghapus angka dengan metode *Regular Expression.* Tahap pra-proses data dapat dilihat pada Kode Sumber 4.13

```
1.  import pandas as pd
2.  data = pd.read_csv('data_ner.csv')
3.
4.  data = data.str.replace('\d+', '')
5.  data = data.str.replace('@"[^\w\s]"',"")
6.  data = data.lower())
```

Kode Sumber 4.13 Pra-Proses Data

4.2.7.2 Pemilihan Fitur berdasar One Hot Encoding

Selanjutnya dilakukan pemilihan fitur untuk dilakukan *clustering.* Pada tugas akhir ini, kolom yang dipilih unbtuk di *cluster* adalah kolom `ner_kategori` yang berisi hasil NER untuk entitas kategori. Dikarenakan kolom tersebut merupakan data nominal, maka harus diubah ke data numerik. Untuk menangani data nominal, bias dengan menggunakan *One Hot Encoding. One Hot Encoding* akan menambah fitur sesuai dengan nama kategori yang ada di fitur `ner_kategori` dan diisi nilai 1 jika pada row tersebut terdapat data kategori yang sesuai. Proses ini dapat dilihat pada Kode Sumber 4.14

```
import pandas as pd
file_text = pd.read_excel('data_ner_cleaned.xlsx'
)
rated_kat = pd.get_dummies(file_text.kategori)
pd.concat([file_text, rated_kat], axis=1)
```

Kode Sumber 4.14 Seleksi Fitur K-Means

4.2.7.3 *K-Means Clustering*

Proses implementasi metode *clustering* dengan *K-Means* dilakukan dengan *library KModes* yang ada pada Python dikarenakan data yang ingin digunakan merupakan data kategorik. Proses clustering dilakukan dalam beberapa jumlah k yaitu k=3 sampai k=10. Implementasi proses clustering dapat dilihat pada Kode Sumber 4.15

```
from kmodes.kmodes import KModes
#Define mnodel KModes
km = KModes(n_clusters=3, init='Huang', n_init=11
, verbose=1)
# fit cluster ke dataframe
clusters = km.fit_predict(rated_kategori)
# mengambil array kmode
kmodes = km.cluster_centroids_
shape = kmodes.shape
# Untuk setiap cluster mode (vector "1" dan "0")

# Mencari and cetak heading kolom ketika "1" munc
ul
# Jika tidak ada "1" masuk ke cluster "no cluster
"
for i in range(shape[0]):
    if sum(kmodes[i,:]) == 0:
        print("\ncluster " + str(i) + ": ")
        print("no cluster")
    else:
        print("\ncluster " + str(i) + ": ")
```

```
        cent = kmodes[i,:]
        for j in rated_kategori.columns[np.nonzero(cent)]:
            print(j)
```

Kode Sumber 4.15 *K-Means Clustering*

4.2.8 Implementasi Klasifikasi *dengan K-Nearest Neighbor (K-NN)*

Pada sub-bab ini akan dijelaskan proses klasifikasi data untuk mengklasifikasikan data lokasi kota yang sudah di deteksi oleh NER ke dalam provinsi sesuai dengan lokasi kota yang sudah terdeteksi pada `ner_lokasi.`

4.2.8.1 Pra-Proses Data

Tahap ini dilakukan pre-processing terhadap data hasil NER dan *clustering* yaitu proses *case folding* , *remove punctuation* , menghapus angka dengan metode *Regular Expression.* Proses ini dapat dilihat pada Kode Sumber 4.16

```
import pandas as pd
data = pd.read_csv('data_ner_clustered.csv')
data = data.str.replace('\d+', '')
data = data.str.replace('@"[^\w\s]"',"")
data = data.lower())
```

Kode Sumber 4.16 Pra-Proses Data

4.2.8.2 Pemilihan Fitur berdasar TF-IDF

Format data dari langkah sebelumnya direpresentasikan sebagai DataFrame dari pustaka pandas, maka pada Kode Sumber 4.17 diubah ke representasi matriks array pustaka numpy untuk mempermudah proses pemilihan fitur.

```
lokasi_kota = pd.DataFrame(texts)['lokasi_extracted'].as_matrix()
```

```
2. lokasi_prov = pd.DataFrame(texts)['lokasi_provins
   i'].as_matrix()
```

Kode Sumber 4.17 Pemilihan Fitur Klasifikasi

Transformasikan teks ke fitur dalam bentuk *vector* yang digunakan sebagai *input estimator* menggunakan fungsi `TfidfVectorizer()` dari pustaka scikit-learn. Pada variable tfidf, atur parameter yang digunakan pada fungsi `TfidfVectorizer()` dengan mengatur sublinear_tf bernilai *True* sehingga rumus IDF yang digunakan memiliki nilai 1 pada numerator dan denumerator untuk menghindari pembagian bernilai 0. Pengaturan fungsi `TfidfVectorizer()` terdapat pada Kode Sumber 4.18

Variabel tv memanggil fungsi `fit_transform()` dengan parameter dari matiks `lokasi_kota`. Maka didapatkan model yang disimpan pada variable tv yang selanjutnya disimpan seperti pada Kode Sumber 4.18 .Selanjutnya kembalikan variable tv menjadi array pada variable features untuk didapatkan nama fiturnya.

```
1. tfidf = TfidfVectorizer(use_idf=True)
2. tv = tfidf.fit_transform(lokasi_kota.astype('U'))

3. features = tv.toarray()
4. lokasi_provinsi = lokasi_prov
5. features.shape
6. featureName=tfidf.get_feature_names()
```

Kode Sumber 4.18 Fungsi TfidfVectorizer

Langkah selanjutnya adalah untuk menggabungkan fitur TF-IDF dengan fitur yang lainnya pada DataFrame `texts`, lalu simpan seluruh fitur pada file berformat csv sebagai backup data. Penggabungan fitur TF-IDF terdapat pada Kode Sumber 4.19

```
for index,ftr in enumerate(featureName):
    texts[ftr]=features[:,index]
texts.to_csv ('/Users/daniseyy/fitur.csv')
```

Kode Sumber 4.19 Menggabungkan Fitur

4.2.8.3 Pemisahan Dataset

Setelah pra-proses dan pemilihan fitur, didapatkan variabel data `texts` yang disimpan dalam bentuk DataFrame. Selanjutnya dilakukan pemisahan dataset untuk memisahkan data *training* sebagai data latih yang digunakan untuk menghasilkan model dan data *testing* untuk menguji hasil model untuk nantinya dijadikan bahan evaluasi.

```
x=texts.drop(columns=['judul', 'konten', 'hasil_ner','ner_lokasi', 'ner_kategori', 'ner_objek', 'kategori_kmode', 'lokasi_extracted', 'lokasi_provinsi','index_doc'])
y=texts['lokasi_provinsi']
```

Kode Sumber 4.20 Pemisahan Dataset

Kode Sumber 4.20 menjelaskan penentuan x sebagai fitur dan y sebagai kelas. Fitur yang diambil terdiri dari lima kolom/fitur seperti yang sudah dijelaskan pada tahap pemilihan fitur berdasarkan TF-IDF.

Pemisahan dataset dilakukan dengan bantuan pustaka sklearn. Pembagian data terbagi menjadi 20% data testing dan 80% data training. Terdapat parameter yang random_state yang diinisiasikan 123. Implementasi pemisahan dataset tertera pada Kode Sumber 4.21.

```
from sklearn.model_selection import train_test_split
```

```
2.  x_train, x_test, y_train, y_test = train_test_spl
    it(x, y, test_size = 0.2, random_state = 123)
```

Kode Sumber 4.21 Pembagian Data *Training & Testing*

4.2.8.4 Normalisasi Data

Pada tahap ini, dilakukan normalisasi fitur yang telah dilakukan pemisahan di tahap sebelumnya. Normalisasi data dilakukan dengan library `StandardScaler` pada pustaka scikit-learn. `StandardScaler` menormalisasi sebuah fitur dengan mengurangi *mean* dan kemudian menskala ke varian unit. Varians unit berarti membagi semua nilai dengan standar deviasi. Implementasi normalisasi data dapat dilihat pada Kode Sumber 4.22.

```
1.  from sklearn.preprocessing import StandardScaler

2.  scaler = StandardScaler()
3.  scaler.fit(x_train)
4.  x_train = scaler.transform(x_train)
5.  x_test = scaler.transform(x_test)
```

Kode Sumber 4.22 Normalisasi Data

4.2.8.5 Klasifikasi dengan K-Nearest Neighbor (K-NN)

Implementasi pengklasifikasian dengan K-Nearest Neighbor (K-NN) dibantu dengan pustaka sklearn dengan jumlah `n_neighbors = 5.` Implementasi klasifikasi dengan KNN dapat dilihat pada Kode Sumber 4.23

```
1.  from sklearn.neighbors import KNeighborsClassifie
    r
2.  klasifikasi = KNeighborsClassifier(n_neighbors=5)

3.
```

```
klasifikasi.fit(x_train, y_train)

y_pred = klasifikasi.predict(x_test)
y_pred

klasifikasi.predict_proba(x_test)
```

Kode Sumber 4.23 Klasifikasi dengan *KNN*

4.2.9 Implementasi Query Hasil dengan MongoDB

Pada proses ini, dilakukan query hasil dari seluruh proses yang sebelumnya. Data hasil proses disimpan di dalam database dengan tools MongoDB. Query yang digunakan adalah untuk menentukan tujuan wisata berdasarkan kategori wisata hasil *clustering* dan lokasi wisata baik secara kota atau provinsi. Implementasi query dapat dilihat pada Kode Sumber 4.24

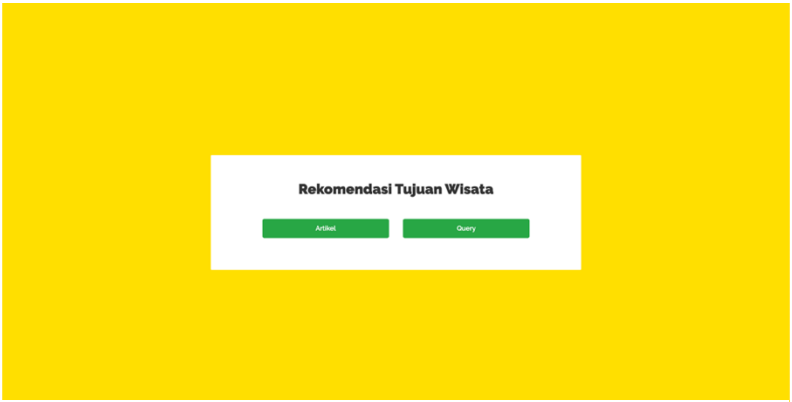
```
from flask import Flask, request, render_template
, make_response
from flask import Markup, flash
from pymongo import MongoClient
connection = MongoClient()
connection = MongoClient('localhost', 27017)
db = connection.data_ta

@app.route('/query/cari', methods=['POST'])
def cari():
    collection = db.data_clustered

    varkat = request.form['kategori']
    varlok = request.form['lokasi']

    myquery = { "$and":[{"kategori_kmode": varkat
},{"$or":[{"lokasi_extracted": varlok},{"lokasi_p
rovinsi": varlok}]}]}


    title = []
    for result in collection.find(myquery):
      title.append(result['judul'])
```

```
21.
22.     return jsonify(result=title)
```

Kode Sumber 4.24 Implementasi Query Hasil

4.3 Implementasi Visualisasi

Implementasi visualisasi dibuat untuk mempermudah uji coba dan evaluasi. Pada halaman *dashboard* aplikasi seperti yang ditunjukkan pada Gambar 4.3 terdapat dua tombol yang akan

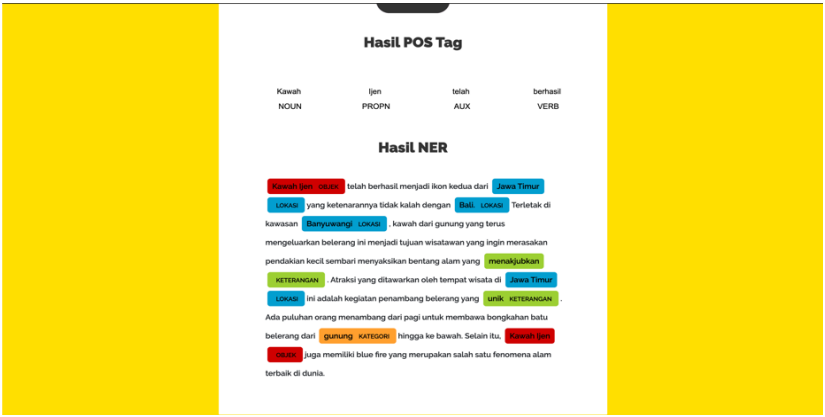


Gambar 4.3 Dashboard Uji Coba Aplikasi

mengarahkan kepada halaman visualisasi yang berbeda, tombol yang pertama adalah untuk *input* data uji coba hasil metode *POS Tag* dan *NER* dan yang kedua adalah untuk menampilkan query hasil *clustering* dan klasifikasi.

Pada halaman *input* data *POS Tag & NER* , pengguna dapat memasukkan teks berupa artikel objek wisata pada *form* seperti ditunjukkan pada Gambar 4.4

Gambar 4.4 Input Data Teks

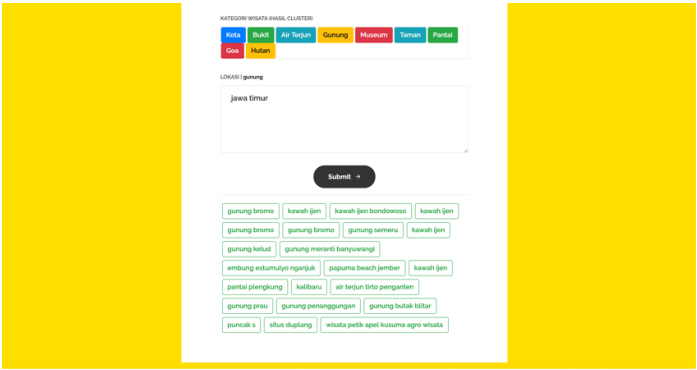


Gambar 4.5 Hasil POS Tag dan NER

Setelah memasukkan teks, pengguna menekan tombol Submit agar system memproses *POS Tagging, Rule-based Matching dan NER* dan menampilkan hasilnya seperti yang ditunjukkan pada Gambar 4.5

Pada halaman query, pengguna dapat memilih kategori wisata yang diinginkan dari hasil *clustering* dan memasukkan lokasi tujuan wisata yang ingin dituju seperti yang ditunjukkan pada Gambar 4.6 baik lokasi secara kota maupun secara provinsi. Setelah memasukkan kata kunci sesuai keinginan, pengguna harus menekan tombol *Submit* agar sistem memproses pencarian. Hasil dari pencarian akan menampilkan objek-objek wisata sesuai dengan pencarian seperti ditunjukkan pada Gambar 4.7

Gambar 4.6 Input Query Hasil Clustering dan Klasifikasi



Gambar 4.7 Hasil Query

BAB V
PENGUJIAN DAN EVALUASI

Pada bab ini akan dijelaskan mengenai rangkaian uji coba dan evaluasi yang dilakukan.

5.1 Lingkungan Pengujian

Lingkungan penguijan sistem pada pengerjaan tugas ini dilakukan pada lingkungan dan alat kakas pada Tabel 5.1 berikut:

Tabel 5.1 Spesifikasi Pengujian

Perangkat	Spesifikasi
Perangkat Keras	• Laptop : Macbook Pro 2018 15-inch • Prosesor: Intel® Core™ i7-7700U CPU @ 2.20GHz (6CPUs), ~3.6GHz • Memori: 256 GB • VGA: Radeon Pro 555x memori GDDR5 4 GB
Perangkat Lunak	• Sistem Operasi macOs Mojave 64-bit • Perangkat Pengembang Visual Studio Code, Sublime Text, Microsoft Word, Jupyter Notebook, Google Chrome, MongoDB Community

5.2 Data Uji Coba

Data yang digunakan untuk evaluasi metode *POS Tag, Named Entity Recognition (NER),* dan *Query* berasal dari data teks artikel online tentang objek wisata di Indonesia.

Data yang digunakan untuk evaluasi metode *clustering* dan klasifikasi pada tugas akhir ini adalah kategori wisata, lokasi hasil *NER* dan lokasi provinsi untuk objek wisata di Indonesia. Dataset yang diambil berjumlah 408 data objek wisata.

Pada pengujian untuk klasifikasi tugas akhir ini, pembagian data dibagi menjadi beberapa metode. Pertama dengan pemisahan *data training* 80% dan *data testing* 20% sehingga didapatkan 312 *data training* dan 96 *data testing*. Untuk prediksi klasifikasi, kita akan mengevaluasi accuracy score, nilai *precision, recall* dan *f-measure* dari setiap nilai n *(neighbor).* Untuk prediksi *clustering,* dilakukan uji evaluasi *K-Means* dengan membandingkan *silhouette score* dari setiap iterasi k.

5.3 Skenario Pengujian 1

5.3.1 Uji Evaluasi Model *Named Entity Recognition (NER)*

Pada tahap ini, dilakukan uji evaluasi untuk model *Named Entity Recognition (NER)* dengan bantuan library `scorer` yang berasal dari pustaka spaCy. Model *Named Entity Recognition (NER)* di evaluasi untuk mendapatkan nilai akurasi, *recall, precision* dan *F-Measure* dari setiap data uji.

5.3.1.1 Data Uji Evaluasi Model *Named Entity Recognition (NER)*

Data yang digunakan pada tahap uji evaluasi ini adalah data teks artikel online tentang objek wisata di Indonesia. Data uji coba dibagi menjadi 3 bagian yang dapat ditinjau dari Tabel 5.2.

Tabel 5.2 Data Uji Evaluasi *NER*

Nama	Jumlah data uji	Keterangan
Data Uji 1	113	Berisi data objek yang sama dengan data training namun berbeda sumber.
Data Uji 2	123	Berisi data objek baru.
Data Uji 3	236	Berisi data gabungan dari Data Uji 1 dan Data Uji 2.

5.3.1.2 Hasil Uji Evaluasi Model *Named Entity Recognition (NER)*

Pada uji evaluasi ini dilakukan uji coba menggunakan library Scorer yang berasal dari pustaka spaCy. Pengujian dilakukan terhadap 3 jenis Data Uji. Hasil dari uji evaluasi model *Named Entity Recognition (NER)* dapat ditinjau pada tabel Tabel 5.3 - Tabel 5.5. dan grafik hasil dari uji evaluasi model dapat dilihat pada Gambar 5.1

Tabel 5.3 Hasil Evaluasi Data Uji 1

Entitas	Hasil Evaluasi			
	Accuracy	Precision	Recall	F-Measure
OBJEK	98,9%	99,8%	91,5%	95,65%
KATEGORI	98,1%	99,5%	100%	99,7%
LOKASI	92,3%	91,5%	89,8%	90,65%
KETERANGAN	98,9%	99,5%	100%	99,75%

Dari hasil evaluasi data pada Data Uji 1 didapatkan bahwa akurasi yang paling tinggi didapatkan oleh entitas OBJEK dan KETERANGAN dengan akurasi 98,9% hal ini dikarenakan pada Data Uji 1 teks yang diuji banyak yang merupakan objek wisata dan keterangan, selanjutnya adalah entitas yang memiliki akurasi terendah adalah entitas LOKASI dengan akurasi 92,3% dikarenakan teks yang diuji tidak membahas tentang lokasi dari objek wisata tersebut.

Tabel 5.4 Hasil Evaluasi Data Uji 2

Entitas	Hasil Evaluasi			
	Accuracy	Precision	Recall	F-Measure
OBJEK	89,5%	89,9%	83,8%	86,85%
KATEGORI	88,8%	87,6%	76,6%	82,1%
LOKASI	84,5%	83,8%	81,9%	82,85%
KETERANGAN	83,1%	82,8%	73,3%	78,05%

Dari hasil evaluasi data pada Data Uji 2 didapatkan bahwa akurasi yang paling tinggi didapatkan oleh entitas OBJEK dengan akurasi 98,9% hal ini dikarenakan pada Data Uji 2 teks yang diuji banyak yang merupakan objek wisata baru yang model belum pernaj kenali, selanjutnya adalah entitas yang memiliki akurasi terendah adalah entitas KETERANGAN dengan akurasi 83,1% dikarenakan teks yang diuji tidak membahas tentang keterangan objek wisata tersebut.

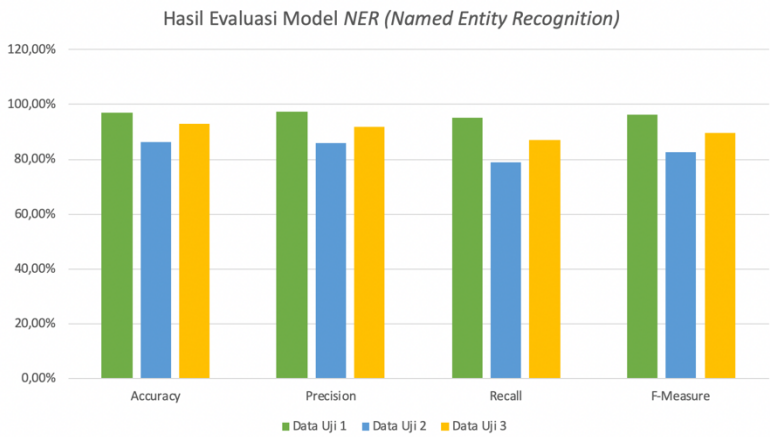
Dari hasil evaluasi data pada Data Uji 2 didapatkan bahwa akurasi yang paling tinggi didapatkan oleh entitas OBJEK dengan

akurasi 98,9% hal ini dikarenakan pada Data Uji 2 teks yang diuji banyak yang merupakan objek wisata baru yang model belum pernaj kenali, selanjutnya adalah entitas yang memiliki akurasi terendah adalah entitas KETERANGAN dengan akurasi 83,1% dikarenakan teks yang diuji tidak membahas tentang keterangan objek wisata tersebut.

Tabel 5.5 Hasil Evaluasi Data Uji 3

Entitas	Hasil Evaluasi			
	Accuracy	Precision	Recall	F-Measure
OBJEK	95,84%	94,85%	87,65%	91,25%
KATEGORI	94,5%	93,55%	88,3%	90,92%
LOKASI	88,9%	87,65%	85,85%	86,75%
KETERANGAN	93,2%	91,15%	86,65%	88,9%

Dari hasil evaluasi dengan 3 data uji bahwa data uji yang memiliki akurasi yang paling baik adalah pada Data Uji 1 dimana data tersebut merupakan data yang serupa dengan data *training* pada model dengan akurasi 97,05%, dengan selanjutnya hasil evaluasi terbaik kedua adalah pada Data Uji 3 dimana data tersebut merupakan gabungan antara data yang serupa dengan data *training* dan data baru yang model belum kenali dengan akurasi sebesar 93,11%, dan hasil evaluasi terakhir yang paling kurang baik adalah pada Data Uji 2 yaitu merupakan data baru yang model belum kenali dengan hasil akurasi sebesar 86,4%.



Gambar 5.1 Grafik Hasil Evaluasi NER

5.4 Skenario Pengujian 2

5.4.1 Uji Evaluasi *Clustering* dengan *K-Means*

Pada sub-bab ini dijelaskan uji coba dan evaluasi hasil clustering dengan metode *K-means.*

5.4.1.1 Uji Evaluasi dengan *Silhouette Score*

Pada tugas akhir ini, evaluasi hasil clustering dilakukan dengan membandingkan *Silhouette Score* dari setiap iterasi k dari mulai k = 3 sampai dengan k=10. Hasil perbandingan *Silhouette Score* dari hasil *clustering* untuk kategori tujuan wisata dapat dilihat pada Tabel 5.6 - Tabel 5.14. Dari hasil perbandingan *Silhouette Score* dapat dilihat bahwa nilai k yang memiliki akurasi terbaik adalah pada k=10 yaitu dengan akurasi skor sebesar 85,8%.

Tabel 5.6 Hasil Uji Silhouette Score

Jumlah k	*Silhouette Score*
3	0,493

Jumlah k	*Silhouette Score*
4	0,568
5	0,628
6	0,638
7	0,699
8	0,734
9	0,838
10	0,858

5.4.1.2 Hasil Clustering sesuai Iterasi k

Sebelumnya telah dilakukan uji evaluasi membandingkan *Silhouette Score* dari setiap iterasi k. Jumlah data yang dihasilkan pada setiap iterasi k dapat dilihat pada

Tabel 5.7 Hasil Uji *Clustering* k=3

No.	*Clusters*	Jumlah Data
0	No - Cluster	225
1	Pantai	38
2	Gunung	140

Tabel 5.8 Hasil Uji Clustering k=4

No.	*Clusters*	Jumlah Data
0	No - Cluster	188
1	Bukit	32
2	Gunung	43
3	Pantai	140

Tabel 5.9 Hasil Uji Clustering k=5

No.	*Clusters*	Jumlah Data
0	No - Cluster	163
1	Gunung	43
2	Pantai	36
3	Air Terjun	19
4	Desa	140

Tabel 5.10 Hasil Uji Clustering k=6

No.	*Clusters*	Jumlah Data
0	No - Cluster	131
1	Gunung	37
2	Air Terjun	32
3	Museum	43
4	Pantai	140
5	Candi	18

Tabel 5.11 Hasil Uji Clustering k=7

No.	*Clusters*	Jumlah Data
0	No - Cluster	139
1	Pantai	17
2	Bukit	43
3	Museum	32
4	Tebing	18
5	Hutan	13
6	Air Terjun	138

Tabel 5.12 Hasil Uji Clustering k=8

No.	*Clusters*	Jumlah Data
0	No - Cluster	127
1	Bukit	32
2	Air Terjun	43
3	Pantai	19
4	Gunung	22
5	Kota	6
6	Museum	13
7	Goa	138

Tabel 5.13 Hasil Uji Clustering k=9

No.	*Clusters*	Jumlah Data
0	No - Cluster	94
1	Air Terjun	37
2	Museum	140
3	Kota	43
4	Taman	8
5	Pantai	18
6	Candi	32
7	Bukit	13
8	Kampung	16

Tabel 5.14 Hasil Uji Clustering k=10

No.	*Clusters*	Jumlah Data
0	No - Cluster	84
1	Gunung	37
2	Bukit	32
3	Museum	19
4	Taman	22
5	Pantai	10
6	Desa	1
7	Kota	13
8	Goa	138
9	Air Terjun	43

5.5 Skenario Pengujian 3

5.5.1 Uji Evaluasi Klasifikasi dengan K-Nearest Neighbor (K-NN)

Pada sub-bab ini dijelaskan uji coba dan evaluasi hasil clustering dengan metode *K-Nearest Neighbor.*

5.5.1.1 Uji Evaluasi dengan *Accuracy Score*

Pada tugas akhir ini, evaluasi hasil clustering dilakukan dengan membandingkan *accuracy score* serta nilai *precision, recall*, dan *f-measure* untuk setiap iterasi n (*neighbor).* Hasil perbandingan accuracy score dari klasifikasi dapat ditinjau pada Tabel 5.15.

Tabel 5.15 Nilai *Accuracy Score* sesuai nilai n

Jumlah n	*Accuracy Score*
3	0,901
4	0,901

Jumlah n	*Accuracy Score*
5	0,901
6	0,851
7	0,827
8	0,827
9	0,827
10	0,827

5.5.2 Hasil Evaluasi Klasifikasi

Sebelumnya telah dilakukan uji evaluasi dari hasil klasifikasi dengan membandingkan dari setiap iterasi n *(neighbor)*. Pada bagian ini merupakan nilai precision, recall dan f-measure dari setiap iterasi n *(neighbor)* yang dapat ditinjau pada Tabel 5.16 - Tabel 5.23.

Tabel 5.16 Hasil untuk n=3

Label Data	*Precision*	*Recall*	*F-Measure*
bali	100%	100%	100%
jawa barat	100%	64%	78%
jawa tengah	73%	95%	83%
jawa timur	98%	96%	97%

Tabel 5.17 Hasil untuk n=4

Label Data	*Precision*	*Recall*	*F-Measure*
bali	100%	100%	100%
jawa barat	100%	64%	78%
jawa tengah	73%	95%	83%
jawa timur	98%	96%	97%

Tabel 5.18 Hasil untuk n=5

Label Data	*Precision*	*Recall*	*F-Measure*
bali	100%	100%	100%
jawa barat	100%	64%	78%
jawa tengah	73%	95%	83%
jawa timur	98%	96%	97%

Tabel 5.19 Hasil untuk n=6

Label Data	*Precision*	*Recall*	*F-Measure*
bali	100%	100%	100%
jawa barat	100%	64%	78%
jawa tengah	62%	100%	77%
jawa timur	100%	84%	92%

Tabel 5.20 Hasil untuk n=7

Label Data	*Precision*	*Recall*	*F-Measure*
bali	100%	100%	100%
jawa barat	100%	50%	67%
jawa tengah	59%	100%	74%
jawa timur	100%	84%	92%

Tabel 5.21 Hasil untuk n=8

Label Data	*Precision*	*Recall*	*F-Measure*
bali	100%	100%	100%
jawa barat	100%	50%	67%
jawa tengah	59%	100%	74%
jawa timur	100%	84%	92%

Tabel 5.22 Hasil untuk n=9

Label Data	*Precision*	*Recall*	*F-Measure*
bali	100%	100%	100%
jawa barat	100%	50%	67%
jawa tengah	59%	100%	74%
jawa timur	100%	84%	92%

Tabel 5.23 Hasil untuk n=10

Label Data	*Precision*	*Recall*	*F-Measure*
bali	100%	100%	100%
jawa barat	100%	50%	67%
jawa tengah	59%	100%	74%
jawa timur	100%	84%	92%

5.6 Skenario Pengujian 4

5.6.1 Uji *Input* Data untuk *POS Tag* dan Named Entity Recognition (*NER)*

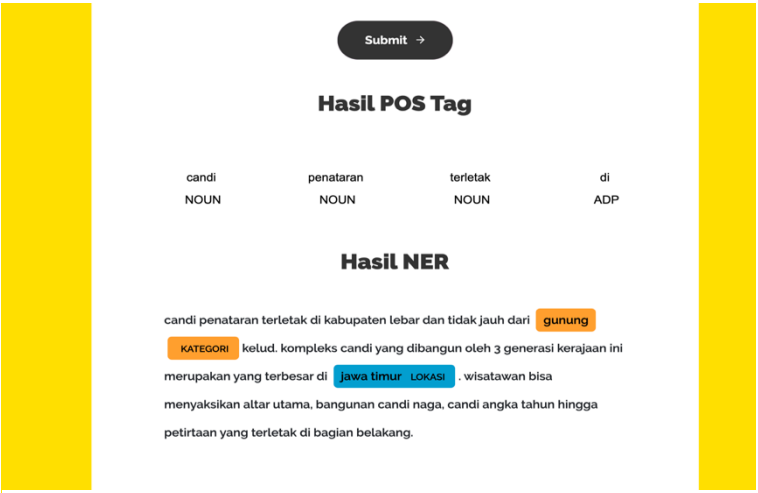
Pada skenario ini dilakukan uji coba input data teks artikel objek wisata menggunakan data masukkan baru ke dalam sistem. Data masukkan baru merupakan data uji coba di luar data latih dan dataset yang sudah ada. Contoh data baru yang akan di uji terdapat pada Tabel 5.24.. Hasil uji coba input data dapat ditinjau pada Gambar 5.2 - 5.6.

Tabel 5.24 Data Uji Input Data

Nama	Data
Data Uji 1	candi penataran terletak di kabupaten lebar dan tidak jauh dari gunung kelud. kompleks candi yang dibangun oleh 3 generasi kerajaan ini merupakan yang terbesar di jawa timur. wisatawan bisa menyaksikan altar utama,

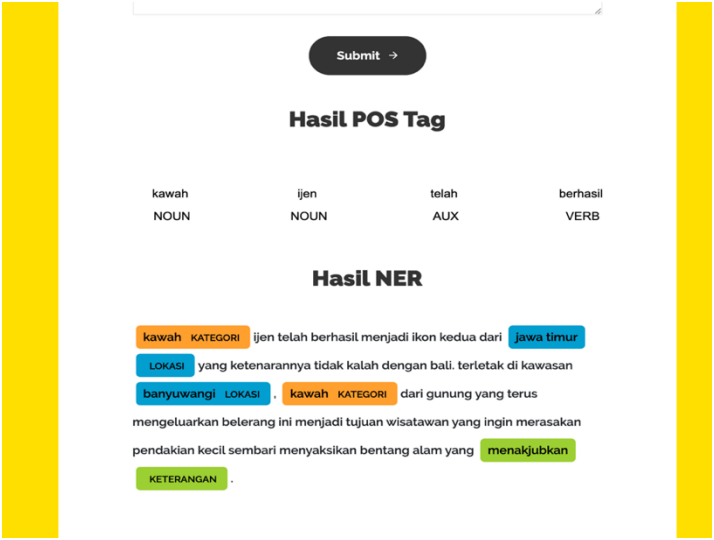
Nama	Data
	bangunan candi naga, candi angka tahun hingga petirtaan yang terletak di bagian belakang.
Data Uji 2	kawah ijen telah berhasil menjadi ikon kedua dari jawa timur yang ketenarannya tidak kalah dengan bali. terletak di kawasan banyuwangi, kawah dari gunung yang terus mengeluarkan belerang ini menjadi tujuan wisatawan yang ingin merasakan pendakian kecil sembari menyaksikan bentang alam yang menakjubkan.
Data Uji 3	Yang paling menarik saat mendatangi pantai sukamade adalah perjalanannya yang cukup menantang. sebelum menuju kawasan mes, wisatawan harus melalui jalanan penuh batu terjal di tengah hutan selama 3-4 jam. setelah sampai di mes, wisatawan akan beristirahat sejenak sebelum penjelajahan dilakukan pada malam hari. berbeda dengan pantai lain yang seru dinikmati saat pagi atau siang, pantai ini justru lebih seru dikunjungi saat malam. wisatawan yang datang ke tempat wisata di jawa timur ini akan dibawa untuk mencari penyu yang akan bertelur di pasir. itulah mengapa wisatawan harus datang saat malam dan mengikuti perintah dari penjaga hutan atau ranger.
Data Uji 4	air terjun coban rondo adalah destinasi unggulan di batu selain aneka taman bermain. terletak di lereng pegunungan, air terjun ini menawarkan 3 pesona kepada pengunjung. pertama tentu saja gerojokan air dari atas bukit yang memberikan

Nama	Data
	efek embun nan segar meski siang hari terasa begitu panas.
Data Uji 5	pantai klayar berlokasi di kecamatan donorojo. tempat wisata alam di jawa timur ini terbilang masih alami, sehingga jalanan menuju pantai ini cukup sulit. pantai klayar adalah salah satu objek wisata unggulan di pacitan. keistimewaan pantai klayar selain pasirnya yang seputih susu, di sini terdapat batu karang yang bentuknya menyerupai spinx dan air mancur alami dengan ketinggian mencapai 10 meter. batuan karang yang ada di pantai klayar ini juga sering disamakan dengan karang di tanah lot, bali.

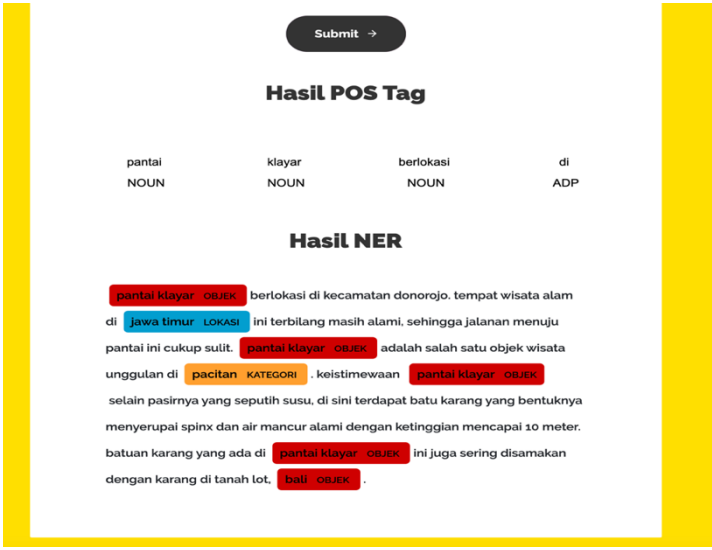


Gambar 5.2 Hasil Data Uji 1

Gambar 5.3 Hasil Data Uji 2



Gambar 5.4 Hasil Data Uji 3



Gambar 5.5 Hasil Data Uji 4

5.6.2 Uji *Query* hasil *clustering* dan klasifikasi

Pada skenario ini dilakukan uji coba *query* label dengan memasukkan kata kunci berupa kategori dan lokasi tujuan wisata baik merupakan kota maupun provinsi. Contoh data uji coba query dapat ditinjau pada Tabel 5.25 Hasil uji coba query dapat dilihat pada Tabel 5.26.

Tabel 5.25 Data Uji Query

Nama	Kategori	*Input* Lokasi	Keterangan
Data Uji 6	Kota	jawa timur	Kata kunci pencarian
	Bukit	jawa timur	
	Air Terjun	jawa timur	

	Gunung	jawa timur	
	Museum	jawa timur	
	Taman	jawa timur	
	Pantai	jawa timur	
	Goa	jawa timur	
	Hutan	jawa timur	

Tabel 5.26 Hasil Query

Nama	**Kategori**	***Input* Lokasi**	**Hasil Q*uery***
Data Uji 6	Kota	jawa timur	Makam ir soekarno
	Bukit	jawa timur	Bukit jamur Bukit jaddih Bukit tinggi daramista Bukit larangan panceng Bukit kolam Bukit sikunir Kawah sikidang Petak dieng Puncak jengger Panorama petung sewu Sendi adventure Bukit semar godang

Nama	Kategori	*Input* Lokasi	Hasil Q*uery*
			Bukit kapur rengel Watu ondo Bukti bunda Gumuk sapu angina Bukit bonsao Kebun the wonosari
	Air Terjun	jawa timur	Air terjun coban rondo Air terjun madakaripura Gili labak madura Air terjun tancak jember Air terjun tancak Air terjun lembah bongok tuban Air terjun lider Air terjun sikarim Air terjun sirawe Curug winong Lubang sewu Sumur jatulanda Coban kembar watu ondo Coban canggu Air terjun surodadu Air terjun banyu langse

Nama	Kategori	*Input* Lokasi	Hasil Q*uery*
			Air terjun sanggrahan Pemandian bektiharjo Air terjun jurug bening Air terjun tirto galuh Air terjun njumeg Air terjun tancak tulis Air terjun antrokan Air terjun maelang Air terjun coban rondo
	Gunung	jawa timur	gunung bromo kawah ijen kawah ijen bondowoso kawah ijen gunung bromo gunung bromo gunung semeru kawah ijen gunung kelud gunung meranti banyuwangi embung estumulyo nganjuk papuma beach jember

Nama	Kategori	*Input* Lokasi	Hasil Q*uery*
			kawah ijen pantai plengkung kalibaru air terjun tirto penganten gunung prau gunung penanggungan gunung butak blitar puncak s situs duplang wisata petik apel kusuma agro wisata
	Museum	jawa timur	museum angkut museum tubuh museum angkut museum angkut kota malang pantai pathok gebang tulungagung museum kailasa dieng plateau theater museum huruf
	Taman	jawa timur	jatim park pemandian taman suruh jatim park jatim park

Nama	Kategori	*Input* Lokasi	Hasil Q*uery*
			batu night spectacular hawai waterpark selecta wisata petik apel eco green park batu night spectacular taman nasional alas purwo penangkaran rusa maliran wana wisata simbad omah kayu batu night spectacular
	Pantai	jawa timur	pantai klayar gili labak gua gong wisata bahari lamongan pantai plengkung gland pantai papuma jember pantai pulau merah banyuwangi pulau sempu malang pantai klayar pacitan pantai gondo mayit blitar

Nama	Kategori	*Input* Lokasi	Hasil Q*uery*
			pantai jonggring saloko malang pantai gua cina malang pantai klayar pacitan pulau bawean gresik teluk biru atau blue bay banyuwangi pulau gili iyang madura pantai jonggring saloko pantai rowo indah pantai klayar pulau bawean blue bay atau teluk biru pulau gili iyang gili labak pantai banyu tibo pulau tabuhan pantai kedung tumpang pantai jonggring saloko malang pulau bawean gresik pantai watu dodol pantai pulau merah

Nama	Kategori	*Input* Lokasi	Hasil Q*uery*
			teluk hijau atau green bay pantai plengkung pulau tabuhan pantai balekambang pantai ngliyep pantai tiga warna pantai balekambang pantai pulau merah gland pantai klayar pantai papuma pulau bawean pulau tabuhan pantai tiga warna house of sampoerna surabaya pantai sanggar tulungagung pantai ngalur tulungagung pantai pulau merah pantai wedi ireng pantai rajegwesi pantai teluk hijau pantai sukamade pantai parang kursi

Nama	Kategori	*Input* Lokasi	Hasil Q*uery*
			pantai bangsring pantai cemara pulau tabuhan pantai banyu tibo pantai ngandul pantai watu karung pantai klayar pantai karang bolong pantai teleng ria pantai srau pantai buyutan pantai pangasan pantai langitan pantai pikatan pantai kunir pantai kasap pantai ngriboyo pantai cemara pantai sowan pantai boom pantai umbul waru pantai peh pulo pantai pangi pantai pasetran gondo mayit pantai jolosutro pantai payangan pantai puger

Nama	Kategori	*Input* Lokasi	Hasil Q*uery*
			pantai papuma pantai paseban pantai balekambang pantai batu bengkung
	Goa	jawa timur	songa rafting probolinggo goa gong goa kancing goa suci goa luweng goa embultuk kasembon rafting
	Hutan	jawa timur	taman nasional baluran taman nasional baluran situbondo ranu kumbolo malang taman nasional meru betiri jemberbanyuwangi taman nasional baluran banyuwangisitubondo taman nasional meru betiri

Nama	Kategori	*Input* Lokasi	Hasil Q*uery*
			taman nasional meru betiri jemberbanyuwangi savana bekol taman nasional baluran jawatan benculuk taman blambangan taman nasional alas purwo

5.7 Skenario Pengujian 5

5.7.1 Uji Evaluasi Perbandingan Input Data dengan Data Uji Manual

Pada uji skenario berikut dilakukan perbandingan input data pada sistem dengan data uji artikel yang di evaluasi secara manual. Data Uji yang digunakan sebagai perbandingan dapat dilihat pada Tabel 5.27. Data Uji tersebut akan dilakukan proses *Named Entity Recognition (NER)* dan akan dilakukan perbandingan dengan deteksi *NER* secara manual.

Tabel 5.27 Data Uji Evaluasi Perbandingan

Nama	Data
Data Uji 1	pantai klayar berlokasi di kecamatan donorojo. tempat wisata alam di jawa timur ini terbilang masih alami, sehingga jalanan menuju pantai ini cukup sulit. pantai klayar adalah salah satu objek wisata unggulan di pacitan. keistimewaan pantai

Nama	Data
	klayar selain pasirnya yang seputih susu, di sini terdapat batu karang yang bentuknya menyerupai spinx dan air mancur alami dengan ketinggian mencapai 10 meter. batuan karang yang ada di pantai klayar ini juga sering disamakan dengan karang di tanah lot, bali.
Data Uji 2	"kota seribu goa" merupakan julukan lain dari kota tuban. hal ini dikarenakan toppers bisa menemukan banyak sekali objek wisata goa yang indah di tuban dan sekitarnya. salah satunya adalah goa putri asih. tampat wisata di tuban satu ini menawarkan pengalaman menjelajahi ekosistem yang sangat eksotis dengan formasi stalaktit dan stalakmit yang indah. lokasi: kawasan hutan jati rph nguluhan, desa nguluhan, kecamatan montong, kabupaten tuban, jawa timur.

5.7.2 Hasil Uji Evaluasi Perbandingan Input Data dengan Data Uji Manual

Uji evaluasi perbandingan input data dengan data uji manual dilakukan dengan melakukan proses *Named Entity Recognition (NER)* pada data uji dan dibandingkan dengan hasil *NER* yang di proses secara manual. Hasil evaluasi perbandingan input data dapat dilihat pada Tabel 5.28.

Tabel 5.28 Hasil Uji Evaluasi Perbandingan

Data	Hasil *NER* (Sistem)	Hasil *NER* (Manual
pantai yang menghadap ke samudra hindia ini memiliki satu hal yang istimewa yaitu gelombang air laut yang bertemu dari arah, selatan, timur dan barat. arus gelombang yang bertabrakan diantara pulau nyonya dan pulau bantengan tersebut menimbulkan efek suara gemuruh yang cukup seru bagi anda yang gemar dengan hal baru.	[(pantai, 'kategori'), (istimewa, 'keterangan'), (pulau, 'objek'), (nyonya, 'objek'), (pulau, 'objek'), (bantengan, 'objek'), (seru, 'keterangan')]	Pantai – Kategori Istimewa – Keterangan Laut – Kategori Pulau Nyonya – Objek Pulau Bantengan – Objek Seru - Keterangan
“kota seribu goa” merupakan julukan lain dari kota tuban. hal ini dikarenakan toppers bisa menemukan banyak sekali objek wisata goa yang indah di tuban dan sekitarnya. salah satunya adalah goa putri asih. tampat wisata di tuban satu ini menawarkan pengalaman menjelajahi ekosistem yang sangat eksotis dengan formasi stalaktit dan stalakmit yang indah. lokasi: kawasan hutan jati rph nguluhan, desa nguluhan, kecamatan	[(kota, 'objek'), (seribu, 'objek'), (goa, 'kategori'), (”, 'keterangan'), (tuban, 'lokasi'), (tuban, 'lokasi'), (goa, 'objek'), (putri, 'objek'), (asih, 'objek'), (tuban, 'lokasi'), (stalaktit, 'keterangan'), (indah,	Kota Seribu Goa – Objek Kota Tuban – Lokasi Indah – Keterangan Tuban – Lokasi Goa Putri Asih – Objek Tuban – Lokasi Eksotis – Keterangan Indah – Keterangan Hutan – Kategori

Data	Hasil *NER* (Sistem)	Hasil *NER* (Manual
montong, kabupaten tuban, jawa timur.	'keterangan'), (hutan, 'objek'), (jati, 'objek'), (desa, 'objek'), (nguluhan, 'objek'), (kecamatan, 'lokasi'), (montong, 'lokasi'), (kabupaten, 'lokasi'), (tuban, 'lokasi'), (jawa, 'lokasi'), (timur, 'lokasi')]	Desa Nguluhan – Lokasi Kecamatan Montong – Lokasi Kabupaten Tuban – Lokasi Jawa Timur - Lokasi

Hasil dari uji evaluasi perbandingan data manual pada data uji 1 dan 2 menghasilkan hasil yang berbeda. Pada Data Uji 1 hasil *NER* yang dihasilkan oleh sistem sudah sesuai dengan uji evaluasi yang dilakukan secara manual, sedangkan pada Data Uji 2 hasil *NER* yang sudah di deteksi oleh sistem memiliki perbedaan dengan hasil *NER* yang dilakukan oleh manual, pada hasil *NER* oleh sistem terdeteksi kata kata seperti “Stalaktit” yang seharusnya tidak terdeteksi oleh sistem namun terdeteksi sebagai entitas Keterangan.

5.8 Evaluasi

Pengujian skenario 1 diperoleh hasil terbaik dari model *Named Entity Recognition (NER)* adalah dengan Data Uji 1 dengan rata-rata nilai *accuracy*, *precision, recall* dan *f-measure* paling terbaik yaitu pada data Uji Coba 1 untuk semua label entitas. Pengujian dilakukan dengan 3 jenis data uji dengan m.

Pengujian skenario 2 diperoleh hasil terbaik *clustering* untuk label kategori adalah dengan jumlah k=10 yang menghasilkan 10 cluster kategori wisata, yaitu Kota, Bukit, Air Terjun, Gunung, Museum, Taman, Pantai, Goa, Hutan dan satu *cluster* tidak bernama. Skor *silhouette score* yang diperoleh adalah 0,85 atau 85%.

Pengujian skenario 3 diperolah hasil terbaik dari klasifikasi data untuk mengklasifikasikan lokasi kota menjadi lokasi provinsi dengan jumlah n=5 dengan evaluasi akurasi *precision, recall* dan *f-measure* paling baik. Skor akurasi yang diperolah yaitu 0,905 atau 90,5%.

Pengujian skenario 4 diperoleh bahwa model dapat mendeteksi POS Tag sesuai dengan sifat kata-kata yang dimiliki dan telah mendeteksi berhasil mendeteksi entitas sesuai label dengan *Named Entity Recognition (NER).* Dalam pengujian *query,* hasil *clustering* dan klasifikasi dapat menampilkan objek-objek wisata yang sesuai dengan kategori dan lokasi sesuai dengan provinsi.

Pengujian skenario 5 diperoleh bahwa dari dua data uji model *Named Entity Recognition (NER)* setelah dibandingkan dengan ekstraksi entitas secara manual menghasilkan hasil ekstraksi yang sama antara sistem dan manual.

[Halaman ini sengaja dikosongkan]

BAB VI
KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan tugas akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang

6.1. Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian yang dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Pengumpulan data untuk *training* dilakukan dengan melakukan *crawling* data dengan bantuan tools *Octoparse*. Data yang sudah di *crawling* selanjutnya dilabeli dan dianotasi dengan bantuan tools anotasi, *Prodigy* untuk mempersiapkan data yang akan digunakan untuk membangun model *Named Entity Recognition (NER).*

2. Membuat model *Name Entity Recognition (NER)* dari bahasa yang belum dimiliki oleh *spaCy* adalah dengan menggunakan model statis Bahasa Indonesia milik spaCy yang dilatih dengan data yang sudah dilakukan anotasi dan dilabeli di tahap sebelumnya. Setelah model awal *NER* sudah jadi, model tersebut dilatih lagi dengan menggunakan data tambahan yang sudah dilabeli dan dianotasi. Semakin banyak *data training* maka model semakin baik dalam mendeteksi kata dan frasa.

3. Membuat model *POS Tagger* memiliki kesamaan tahapan dengan pembangunan model *Name Entity Recognition.* Pembangunan model dengan menggunakan *library SpaCy*. Model *POS Tagger* digunakan untuk melakukan *rule-based matching.*

Penggunakan *rule-based matching* bertujuan untuk mendeteksi kata dan frasa yang tidak terdeteksi oleh proses *NER*.

4. Penggunaan metode *clustering k-means* untuk meng-*cluster* data dapat dapat membantu meng-*cluster* data yang berbentuk kategorik seperti pada tugas akhir ini dengan jumlah k yang optimal adalah 10 dan skor akurasi yang diperoleh sebesar 0,858 atau 85,8%. Penggunaan metode klasifikasi *K-Nearest Neighbor* untuk membantu mengklasifikasi data entitas lokasi dengan jumlah n (neighbor) yang optimal berjumlah 5 diperoleh skor akurasi sebesar 0,901 atau 90,1%.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi, dan pengujian yang telah dilakukan. Di antaranya adalah sebagai berikut:

1. Menambah entitas label yang di deteksi oleh *Named Entity Recognizer (NER)* untuk mendeteksi lebih banyak entitas.
2. Melakukan uji coba mengolah data entitas label dengan metode clustering lainnya seperti *Mixture Modelling* atau *Suffix Tree Clustering,* sehingga dapat dilakukan perbandingan dan perbedaan hasil setiap penggunaan metode *clustering*.
3. Melakukan uji coba mengolah data entitas label dengan metode klasifikasi lainnya seperti *Naïve Bayes*, *Random Forest*, dan *Support Vector Machine (SVM),* sehingga dapat dilakukan perbandingan dan perbedaan hasil setiap penggunaan metode clustering.

DAFTAR PUSTAKA

[1] J. Li, L. Xu, L. Tang, S. Wang, and L. Li, "Big data in tourism research: A literature review," *Tour. Manag.*, vol. 68, pp. 301–323, 2018, doi: 10.1016/j.tourman.2018.03.009.

[2] "Rekomendasi Pariwisata di Indonesia." [Online]. Available: https://elib.unikom.ac.id/files/disk1/584/jbptunikompp-gdl-herdiansya-29154-8-unikom_h-i.pdf. [Accessed: 27-Dec-2019].

[3] "Mengenal Apa itu Pengertian Web Crawler - Kamus Hosting IDCloudHost." [Online]. Available: https://idcloudhost.com/kamus-hosting/web-crawler/. [Accessed: 27-Dec-2019].

[4] P. Pengembangan and A. Dan, "Bab I," pp. 1–31, 1994.

[5] K. Widhiyanti and A. Harjoko, "POS Tagging Bahasa Indonesia Dengan HMM dan Rule Based," *J. Inform.*, vol. 8, no. 2, 2013, doi: 10.21460/inf.2012.82.125.

[6] "Ekstraksi Informasi: Named Entity Recognition – Blog Yudi Wibisono." [Online]. Available: https://yudiwbs.wordpress.com/2012/02/07/named-entity-recognition/. [Accessed: 27-Dec-2019].

[7] "Rule-Based Matching with spaCy - Ashiq KS - Medium." [Online]. Available: https://medium.com/@ashiqgiga07/rule-based-matching-with-spacy-295b76ca2b68. [Accessed: 27-Dec-2019].

[8] “spaCy 101: Everything you need to know · spaCy Usage Documentation.” [Online]. Available: https://spacy.io/usage/spacy-101. [Accessed: 26-Dec-2019].

[9] “Flask - Wikipedia bahasa Indonesia, ensiklopedia bebas.” [Online]. Available: https://id.wikipedia.org/wiki/Flask. [Accessed: 27-Dec-2019].

[10] B. a B. Ii and a P. Sistem, “Universitas Sumatera Utara 7,” pp. 7–37, 2001.

[11] “GitHub - famrashel/idn-tagged-corpus: Indonesian Manually Tagged Corpus.” [Online]. Available: https://github.com/famrashel/idn-tagged-corpus. [Accessed: 09-Jan-2020].

BIODATA PENULIS

Denise Sonia Rahmadina, lahir di Jakarta pada tanggal 9 Agustus 1998. Penulis menempuh Pendidikan mulai SD Islam Terpadu Fajar Hidayah (2004-2008), SD Negeri Kelapa Dua Wetan 01 Pagi Jakarta (2008-2010), SMP Negeri 19 Jakarta (2010-2013), SMA Negeri 28 Jakarta (2013-2016) dan sekarang sedang menempuh Pendidikan S1 Informatika di ITS. Penulis aktif dalam organisasi dan kepanitiaan Himpunan Mahasiswa Teknik Computer (HMTC) dan aktif dalam kepanitiaan Schematics ITS. Diantaranya adalah menjadi staff Hubungan Luar HMTC 2017-2018, BPH III 3D (Desain, Dekorasi dan Dokumentasi) Schematics 2017, serta menjadi Wakil Kepala Departemen Hubungan Luar HMTC 2018-2019. Penulis berpengalaman sebagai asisten dosen pada matakuliah Sistem Basis Data, Manajemen Basis Data dan Manajemen Proyek Perangkat Lunak dan pernah melaksanakan kerja praktik sebagai *Data Analyst* di PT. Tokopedia di Kuningan, Jakarta.