TUGAS AKHIR – IF184802

IMPLEMENTASI WEB SISTEM INFORMASI UNTUK VISUALISASI RIWAYAT BERITA ONLINE TOPIK COVID-19 DI INDONESIA

MUHAMMAD NAUFAL REFADI
05111740000097

Dosen Pembimbing I
Diana Purwitasari, S.Kom., M.Sc.

Dosen Pembimbing II
Agus Budi Raharjo, PhD

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya
2021

[*Halaman ini sengaja dikosongkan*]

TUGAS AKHIR – IF184802

IMPLEMENTASI WEB SISTEM INFORMASI UNTUK VISUALISASI RIWAYAT BERITA ONLINE TOPIK COVID-19 DI INDONESIA

MUHAMMAD NAUFAL REFADI
05111740000097

Dosen Pembimbing I
Diana Purwitasari, S.Kom., M.Sc.

Dosen Pembimbing II
Agus Budi Raharjo, PhD

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya
2021

TUGAS AKHIR – IF184802

IMPLEMENTASI WEB SISTEM INFORMASI UNTUK VISUALISASI RIWAYAT BERITA ONLINE TOPIK COVID-19 DI INDONESIA

MUHAMMAD NAUFAL REFADI
05111740000097

Dosen Pembimbing I
Diana Purwitasari, S.Kom., M.Sc.

Dosen Pembimbing II
Agus Budi Raharjo, PhD

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya
2021

[*Halaman ini sengaja dikosongkan*]

UNDERGRADUATE THESIS – IF184802

IMPLEMENTAION OF A WEB INFORMATION SYSTEM FOR VISUALIZING ONLINE NEWS HISTORY ON COVID-19 TOPICS IN INDONESIA

Muhammad Naufal Refadi
05111740000097

Supervisor I
Diana Purwitasari, S.Kom., M.Sc.

Supervisor II
Agus Budi Raharjo, PhD

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

[*Halaman ini sengaja dikosongkan*]

LEMBAR PENGESAHAN

IMPLEMENTASI WEB SISTEM INFORMASI UNTUK VISUALISASI RIWAYAT BERITA ONLINE TOPIK COVID-19 DI INDONESIA

TUGAS AKHIR

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

Oleh:
Muhammad Naufal Refadi
NRP: 05111740000097

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Diana Purwitasari, S.Kom., M.Sc.
NIP. 197804102003122001

(Pembimbing 1)

Agus Budi Raharjo, PhD
NIP. 1990202011022

(Pembimbing 2)

SURABAYA
FEBRUARI 2021

[*Halaman ini sengaja dikosongkan*]

IMPLEMENTASI WEB SISTEM INFORMASI UNTUK VISUALISASI RIWAYAT BERITA ONLINE TOPIK COVID-19 DI INDONESIA

Nama Mahasiswa : Muhammad Naufal Refadi
NRP : 05111740000097
Departemen : Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, ITS
Dosen Pembimbing 1 : Diana Purwitasari, S.Kom., M.Sc.
Dosen Pembimbing 2 : Agus Budi Raharjo, PhD

Abstrak

Seluruh dunia saat ini sedang mengalami sebuah virus yang mematikan dan memiliki tingkat penyebaran yang sangat cepat bernama Coronavirus Disease of 2019 atau bisa disebut COVID-19. Virus ini telah menyebar dan memakan korban jiwa di seluruh negara termasuk Indonesia. Di Indonesia sendiri kasus pertama muncul pada 2 Maret dan sampai sekarang pada 30 Juny 2021, jumlah kasus virus COVID-19 terus meningkat. Banyak kejadian atau hal yang menarik yang dapat menggambarkan situasi pandemi di Indonesia seperti usaha pemerintah pusat atau daerah dalam mengatasi kenaikan kasus, situasi masyarakat saat diberlakukannya protokol kesehatan dan lain-lain. Dari kejadian itulah bisa digambarkan menjadi sebuah visualisasi yang dapat menggambarkan perjalanan COVID-19 di Indonesia.

Buku ini membahas tentang visualisasi berita tentang COVID-19 di Indonesia yang diambil dari portal berita online. Dipilihnya portal berita online karena selain sebuah media yang bisa menggambarkan situasi yang terjadi di Indonesia, juga portal media online memiliki data berita yang besar. Data berita tersebut akan diambil dengan melakukan web scraping dan akan diolah agar mendapatkan informasi sesuai dengan tema COVID-19 di Indonesia. Data yang telah diolah nantinya akan dilakukan

pemrosesan teks dan klasifikasi teks untuk mencari dan memvisualisasikan topik berita yang terpopuler dari setiap tanggal. Hasil data tersebut nantinya akan divisualisasikan menjadi sebuah graph dan chart yang dapat membantu hasil analisa..

Kata kunci: COVID-19, Indonesia, Portal Berita Online, Berita, Web Scraping, Visualisasi, Pemrosesan Teks,Klasifikasi Teks

IMPLEMENTAION OF A WEB INFORMATION SYSTEM FOR VISUALIZING ONLINE NEWS HISTORY ON COVID-19 TOPICS IN INDONESIA

Student Name	: Muhammad Naufal Refadi
Registration Number	: 05111740000097
Department	: Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, ITS
First Supervisor	: Diana Purwitasari, S.Kom., M.Sc.
Second Supervisor	: Agus Budi Raharjo, PhD

Abstract

The whole world is currently experiencing a virus that is deadly and has a very fast spreading rate called Coronavirus Disease of 2019 or it can be called COVID-19. This virus has spread and claimed many lives in all countries including Indonesia. In Indonesia alone, the first cases appeared on March 2 and until now , Juny 27 2021, the number of cases of the COVID-19 virus continues to increase. There are many interesting events or things that can describe the pandemic situation in Indonesia, such as the efforts of the central or local government to deal with the increase in cases, the situation in the society when the health protocol is implemented and many others. From these incident, it can be visualized to describe the journey of COVID-19 in Indonesia.

This book discusses the visualization of news about COVID-19 in Indonesia which is taken from an online news portal. The online news portal was chosen because in addition to being a media that can describe the situation in Indonesia, online media portals also have large news data. The news data will be retrieved by doing web scraping and will be processed to match the COVID-19 theme in Indonesia. The news data will be retrieved by doing web scraping and will be processed to get information about

COVID-19 theme in Indonesia. The data that has been processed will then be text processed and classificated to find and visualize the most popular news topics from each date and place.

Keywords: COVID-19, Indonesia, Online News Portal, News , Web Scraping, Visualization, Text Processing, Text Clasisification

KATA PENGANTAR

Segala puji syukur bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dan laporan akhir dalam bentuk buku ini yang berjudul:

IMPLEMENTASI WEB SISTEM INFORMASI UNTUK VISUALISASI RIWAYAT BERITA ONLINE TOPIK COVID-19 DI INDONESIA

Pengerjaan tugas akhir ini dilakukan untuk memenuhi salah satu syarat meraih gelar Sarjana di Departemen Teknik Informatika Fakultas Teknologi Elektro dan Informatika Cerdas Institut Teknologi Sepuluh Nopember.

Dengan selesainya tugas akhir ini diharapkan apa yang telah dikerjakan penulis dapat memberikan manfaat bagi perkembangan ilmu pengetahuan terutama di bidang teknologi informasi serta bagi diri penulis sendiri.

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan baik secara langsung maupun tidak langsung selama pengerjaan tugas akhir maupun selama masa studi antara lain:

1. Terimakasih kepada Allah SWT, di mana penulis masih diberi kesempatan, kesehatan, dan umur untuk menempuh kuliah di sini dan menjalani hidup dengan baik.
2. Kedua orang tua, Budiyono dan Erfanti Qodarsih, serta keluarga penulis yang senantiasa mendukung, mendoakan serta memotivasi penulis dalam menyelesaikan Tugas akhir ini.
3. Ibu Diana Purwitasari, S.Kom., M.Sc. sebagai Dosen Pembimbing I dan Ibu Agus Budi Raharjo, PhD. sebagai Dosen Pembimbing II penulis yang senantiasa membimbing, memberikan ilmu, memberikan arahan,

memberikan pendapat serta bantuan-bantuan lainnya dalam pengerjaan Tugas Akhir ini.

4. Zahrul Zizki Dinanto dan Isnaini Nurul Kurniasari sebagai rekan dalam pengerjaan Tugas Akhir ini.
5. Seluruh dosen dan karyawan Departemen Teknik Informatika yang telah memberikan ilmu dan pengalaman kepada penulis selama masa kuliah di Teknik Informatika ITS
6. Teman-teman Teknik Informatika ITS, khususnya angkatan 2017, yang senantiasa membantu dan menemani perjalanan kuliah penulis sampai pada tahap terakhir ini.
7. Komunitas *open source*, dan forum Stack Overflow yang membantu penulis dalam mendapatkan jawaban atas kendala selama pengerjaan Tugas Akhir ini.
8. Responden yang bersedia mengisi survei mengenai sistem informasi.
9. Serta pihak-pihak lain yang tidak dapat disebutkan di sini yang telah banyak membantu penulis dalam penyusunan tugas akhir ini.

Penulis menyadari bahwa buku ini jauh dari kata sempurna. Maka dari itu, penulis mohon maaf apabila masih ada kekurangan pada tugas akhir ini. Penulis juga mengharapkan kritik dan saran yang membangun untuk pembelajaran dan perbaikan di kemudian hari. Penulis berharap buku ini dapat berkonstribusi dalam ilmu pengetahuan dan manfaat yang sebaik-baiknya.

Surabaya, Juli 2021

Muhammad Naufal Refadi

[*Halaman ini sengaja dikosongkan*]

DAFTAR ISI

[*Halaman ini sengaja dikosongkan*]

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR KODE SUMBER

[*Halaman ini sengaja dikosongkan*]

BAB I
PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan masalah, batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan buku tugas akhir ini.

1.1 Latar Belakang

Pada akhir 2019, dunia dilanda dengan wabah virus mematikan bernama Coronavirus Disease of 2019 atau bisa disebut COVID-19. COVID-19 adalah penyakit yang disebabkan oleh virus severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) dan merupakan penyebab dari penyakit fatal yang menjadi perhatian besar kesehatan masyarakat global [1]. Virus ini pertama kali ditemukan di kota Wuhan, Cina pada Desember 2019. Virus COVID-19 ini menyerang sistem pernapasan dan memiliki gejala umum seperti batuk, panas dan sesak nafas [2]. Tercatat pada 27 Januari 2020, sudah 100 juta kasus infeksi COVID-19 di seluruh dunia dengan diantaranya 2,1 juta jiwa meninggal dunia karena virus tersebut termasuk Indonesia [3].

Di Indonesia sendiri, kasus COVID-19 pertama dan kedua di Indonesia terdeteksi pada tanggal 2 Maret 2020 [4]. Tercatat sampai 27 Januari, sudah lebih dari 1 juta kasus di Indonesia dengan diantaranya 28.468 meninggal karena virus tersebut [5]. Banyak upaya yang dilakukan pemerintah untuk mengurangi kenaikan kasus COVID-19 dari Pembatasan Sosial Berskala Besar (PSBB) sampai vaksinasi .

Pada masa pandemi COVID-19 ini, banyak sekali visualisasi tentang COVID-19 seperti pertumbuhan kasus di berbagai daerah. Selain sebagai sumber informasi kepada masyarakat, visualisasi ini dapat membantu pemerintah, organisasi kesehatan dan masyarakat agar bisa menjadi patokan dalam program mengurangi kenaikan angka kasus COVID-19, atau kasus virus yang baru kemudian harinya. Maka dari itu

diperlukan gambaran Banyak cara untuk menggambarkan perjalanan COVID-19 di Indonesia, tetapi pada tugas akhir ini penulis akan melakukan visualisasi melalui portal berita online.

Selama pandemi COVID-19, Media berita selalu memiliki peranan penting dalam menginformasikan berita kepada pembaca untuk memuaskan keingintahuan mereka dalam menghadapi wabah COVID-19 [6]. Selain itu pada portal berita online memiliki kemudahan akses dan jumlah data berita yang besar dalam berbagai bidang. Data-data tersebut dapat dianalisis dan diklasifikasikan untuk mendapatkan informasi terutama mengenai COVID-19 di Indonesia [7].

Dari Informasi diatas, salah satu penyelesaiannya adalah membuat sebuah visualisasi berita mengenai riwayat perjalanan COVID-19 di setiap daerah provinsi di Indonesia menggunakan berita yang diambil dari portal berita online. Untuk memberikan informasi yang jelas dan efektif, berita yang diambil berasal dari portal berita terpecaya yaitu Tribunnews.com, Kompas.com. karena memiliki jumlah pengunjung sangat banyak dan terbaik [8]. Di dalam tugas akhir ini, penulis mengusulkan membuat sebuah sistem informasi web visualisasi Riwayat Berita Online. Sistem informasi ini merupakan sebuah web yang menampilkan visualisasi riwayat perjalanan COVID-19 di setiap provinsi di Indonesia melalui berita dari portal berita online menggunakan teknik web scraping. Berita yang ditampilkan harus berhubungan dengan COVID-19 dan berada pada sesuai daerah provinsi masing-masing. Selain itu penulis juga menambah visualisasi informasi yang tidak ada dalam artikel berita untuk membantu mendapatkan hasil analisia berita yang didapatkan melalui pemrosesan teks dan klasifikasi teks pada data berita yang telah didapatkan. Diharapkan, visualisasi ini dapat membantu menginformasikan masyarakat dan membantu pemerintah pusat dan daerah dalam menentukan kebijakan dalam mengurangi angka kenaikan kasus COVID-19.

1.2 Rumusan Masalah

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

1. Bagaimana melakukan pengumpulan data dari portal berita online?
2. Bagaimna melakukan proses klasifikasi teks dalam mendapatkan informasi baru?
3. Bagaimana menampilkan visualisasi Riwayat Berita Online Topik COVID-19?
4. Bagaimana melakukan implementasi Sistem Informasi untuk Visualisasi Riwayat Berita Online ke dalam sebuah web?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Dataset yang digunakan berasal dari KawalCOVID19 berupa Total kasus, jumlah kasus setiap hari, sembuh setiap hari,kematian setiap hari di seluruh provinsi di Indonesia pada kurun waktu 18 Maret 2020 - Juli 2021.
2. Website yang akan diolah berasal dari Tribunnews.com dan Kompas.com yang diambil tanggal Maret 2020-Juli 2021.
3. Pengambilan data berita yang diambil menggunakan Python dengan *framework* Scrapy.
4. Model yang digunakan dalam proses klasifikasi teks adalah *Random Forest, Support Vector Machine, K-Nearest Neighbor, Multinomial Naïve Bayes*.
5. Sistem informasi yang dibangun menggunakan Bahasa Pemrograman Web (PHP, HTML, CSS, JavaScript) dengan *framework* laravel dan MYSQL sebagai *database*-nya.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah untuk membangun sebuah website sistem informasi yang menampilkan visualisasi riwayat berita online topik COVID-19 dari awal kasus COVID-19 di Indonesia.

1.5 Manfaat

Manfaat Tugas Akhir ini adalah sebagai berikut:

1. Dapat membangun sebuah website sistem informasi visualisasi yang dapat memudahkan masyarakat Indonesia melihat dan membandingkan pola perkembangan kasus COVID-19 di setiap provinsi di seluruh Indonesia.
2. Dapat menjadikan bahan visualisasi dalam membantu program pemerintah pusat dan daerah dalam menekan kasus COVID-19 dan kasus virus lainnya di kemudian hari

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan tugas akhir ini yaitu:

1.6.1 Penyusunan Proposal Tugas Akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan, tinjauan pustaka, metodologi dan jadwal kegiatan dari tugas akhir yang akan dibuat. Pendahuluan terdiri atas hal yang menjadi latar belakang pada usulan tugas akhir ini, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil tugas akhir. Dijabarkan juga tinjauan pustaka sebagai bahan referensi dan pendukung dalam pembuatan tugas akhir. Metodologi yang berisi penjelasan mengenai tahapan penyusunan tugas akhir. Dan yang terakhir jadwal kegiatan dalam pengerjaan tugas akhir.bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

1.6.2 Studi Literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi berupa artikel,paper dan dokumentasi yang diperlukan dalam pembuatan sistem informasi website yaitu mengenai scraping dari portal berita, visualisasi data, pembuatan website visualisasi dan topik-topik lainnya yang berkaitan dengan pengerjaan tugas akhir ini.

1.6.3 Analisis dan Perancangan Sistem

Pada tahap ini, analisis kebutuhan dan perancangan sistem dilakukan untuk untuk merumuskan solusi yang tepat dalam melakukan analisa data, visualisasi dan mengimplementasikannya ke dalam sebuah sistem informasi web. Tahap desain meliputi arsitektur perangkat lunak yang digunakan, desain visualisasi, desain antarmuka, serta fitur-fitur yang mendukung ke dalam sebuah sistem informasi web.

1.6.4 Implementasi

Pada tugas akhir ini, proses scraping, pemrosesan teks dan visualisasi menggunakan bahasa pemrograman Python dengan library sesuai kebutuhan masing-masing. Dan Sistem informasi ini menggunakan bahasa pemrograman berbasis web (PHP, HTML, CSS, JS) dengan kerangka kerja framework Laravel dan MySQL sebagai database..

1.6.5 Pengujian dan Evaluasi

Tahap pengujian dan evaluasi merupakan salah satu tahapan yang digunakan untuk mengetahui performa dari hasil implementasi visualisasi dan sistem informasi. Pengujian dan evaluasi akan dilakukan sebagai berikut :

1. Pengujian dan evaluasi kesesuaian atribut label pada dataset berita yang didapatkan dari proses klasifikasi teks.
2. Pengujian usability dilakukan dengan cara melakukan survei ke pengguna untuk mengukur tingkat kegunaan dari sistem informasi visualisasi yang telah dibuat untuk membantu pengguna.

1.6.6 Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi yang telah dibuat.

Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
   a. Latar Belakang
   b. Rumusan Masalah
   c. Batasan Masalah
   d. Tujuan
   e. Manfaat
   f. Metodologi
   g. Sistematika Penulisan
2. Dasar Teori
3. Analisis dan Perancangan Sistem
4. Implementasi
5. Uji Coba dan Evaluasi
6. Kesimpulan dan Saran
7. Daftar Pustaka

1.7 Sistematika Penulisan

Buku tugas akhir ini merupakan laporan secara lengkap mengenai tugas akhir yang telah dikerjakan baik dari sisi teori, analisis, rancangan, maupun implementasi sampai uji coba dan evaluasi, sehingga memudahkan bagi pembaca dan juga pihak yang ingin mengembangkannya lebih lanjut. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

Bab I Pendahuluan

Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini berisikan tentang hal-hal berupa literatur dan pustaka-pustaka yang menunjang dan berhubungan dengan pengerjaan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini menjelaskan mengenai rancangan desain sistem yang akan dibangun. Mulai dari desain proses pengumpulan data, desain pemrosesan teks , desain proses klasifikasi teks, dan desain visualisasi web.

Bab IV Implementasi

Bab ini menjelaskan mengenai bagaimana implementasi yang dilakukan dari analisis dan desain sistem yang sudah dirancang. Implementasi ini dilakukan menggunakan bahasa pemrograman Python dan framework Laravel.

Bab V Uji Coba dan Evaluasi

Bab ini menjelaskan terkait metode pengujian yang digunakan dan membahas mengenai bagaimana pengujian dilakukan pada setiap tahap pembuatan sistem dan hasil pengujian tersebut. Serta untuk mengetahui kesesuaian metode pengujian yan g dilakukan dengan data-data yang telah terkumpul.

Bab VI Kesimpulan dan Saran

Bab ini menjelaskan tentang kesimpulan dari hasil pengujian yang telah dilakukan serta saran-saran yang dapat digunakan untuk pengembangan sistem selanjutnya.

[*Halaman ini sengaja dikosongkan*]

BAB II
DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori penunjang dalam melakukan implementasi web sistem informasi untuk visualisasi riwayat berita online topik COVID-19 di Indonesia. Di antaranya meliputi web scraping untuk pengumpulan data, pemrosesan teks dan visualisasi untuk pemilihan kata kunci, klasifikasi teks, teknologi web untuk visualisasi riwayat berita online.

2.1 Web Scraping

Web scraping atau *web extracting* merupakan sebuah teknik untuk mengekstrak data spesifik dari *World Wide Web* (WWW) dan disimpan ke dalam berkas sistem atau database untuk pengambilan keputusan atau analisis selanjutnya [9]. Pada tugas akhir ini, proses web scraping akan menggunakan bahasa pemrograman Python dan menggunakan framework Scrapy. Scrapy merupakan sebuah framework yang cocok untuk melakukan web scraping terutama pada web yang terdiri dari banyak struktur halaman yang sama. Berikut merupakan langkah-langkah dalam teknik web scraping menggunakan framework Scrapy [10]:

a. Pengguna menentukan halaman yang akan di-scrape dan menghasilkan template untuk konten yang diinginkan
b. pengguna memilih sekumpulan tautan yang mengarah ke halaman yang cocok dengan template konten yang ditentukan oleh pengguna
c. Pengguna menentukan format output data.
d. Scrapy melakukan crawl dari tautan yang dipilih oleh pengguna dan scrape konten sesuai template pengguna.

2.2 Pemrosesan dan Visualisasi Teks

Pemrosesan Teks merupakan tahapan melakukan seleksi dan normalisasi kata menjadi menjadi bentuk perantara yang nantinya akan dilakukan berbagai teknik dalam *text mining* [11]. Dalam pemrosesan teks nantinya akan dilakukan proses lemmatilisasi, menghapus tanda baca, menghapus stopwords dan menghitung jumlah frekuensi tiap data.

Pada buku ini, penulis akan melakukan visualisasi dari data hasil pemrosesan teks melalui *bar chart* dan *stacked bar chart* , untuk mendapatkan sebuah informasi yang baru. *Bar chart* adalah diagram batang yang bukan hanya membandingkan data pada tiap kategori [12]. Sedangkan *stacked bar chart* adalah diagram batang yang bukan hanya membandingkan data pada tiap kategori, tetapi juga kemampuan memecah dan membandingkan nilai dari setiap atribut data yang berkontribusi dalam nilai total data [13].

2.3 Klasifikasi Teks

Klasifikasi teks atau kategorisasi teks merupakan proses otomatis yang menempatkan teks dokumen ke dalam suatu kategori berdasarkan isi dari teks tersebut. Proses klasifikasi teks dimulai dari persiapan data, pemrosesan teks, pemilihan fitur/rekayasa fitur, pelatihan model sampai pemberian kategori [14]. Pada tugas akhir ini proses klasifikasi teks digunakan pada data berita untuk mendapatkan informasi yang tidak ada data berita yaitu label berita.

Tahap awal dalam proses klasifikasi teks, menentukan kategori label yang akan digunakan untuk mengelompokkan setiap teks data kedalam kategori label tersebut. Setelah itu, tentukan sampel dataset teks yang setiap datanya mempunyai label yang telah ditentukan sebelumnya.

Tahap kedua adalah pemrosesan teks. Tahap ini mengubah teks menjadi bentuk yang lebih mudah dimengerti sehingga tidak ada distorsi yang diperkenalkan ke model dan algoritma

pembelajaran mesin dapat bekerja lebih baik [14]. Berikut tahap pemrosesan teks pada tugas akhir ini:

1. Menghapus karakter spesial seperti “\n”,”\t”. Lalu mengubah semua huruf besar menjadi kecil.
2. Mengubah huruf besar menjadi huruf kecil
3. Menghapus tanda baca seperti “.”, “!“, “-“
4. Merubah kata infleksi menjadi kata dasar atau bias disebut letimasi teks
5. Menghapus beberapa kata yang tidak penting atau *stopword*

Tahap ketiga adalah rekayasa fitur. Rekayasa fitur adalah sebuah proses mengubah data menjadi fitur untuk model pembelajaran mesin sehingga fitur berkualitas baik dalam membantu meningkatkan kinerja model [15]. Penulis menggunakan fitur TF-IDF atau *Term Frequency – Inverse Document Frequency* dalam menjalankan tahap rekayasa fitur. TF-IDF adalah bobot dari sejumlah kata dalam dokumen dalam seluruh korpus. Formula dalam proses pembobotan TF-IDF dapat dilihat pada Persamaan 2.1.

$$TFIDF(t,d) = TF(t,d) \times \log(\frac{N}{DF(t)}) \quad (2.1)$$

Persamaan 2.1 Perhitungan bobot kata menggunakan metode TF-IDF

Keterangan :

- t = sebuah kata dalam satu dokumen
- d = dokumen
- tf(t) = frekuensi kata yang muncul di dokumen d
- N = jumlah dokumen di dalam korpus
- DF(t) = jumlah dokumen didalam korpus

Nilai bobot dari TFIDF selalu meningkat proporsional berdasarkan jumlah kata yang muncul dalam dokumen dan diimbangi den- gan banyak dokumen dalam korpus yang berisi kata tersebut.

Tahap terakhir adalah pelatihan model, tahap ini melakukan uji performa data latih pada beberapa jenis model. Model yang memiliki akurasi data latih paling tinggi digunakan sebagai model utama dalam proses klasifikasi teks.

Berikut merupakan jenis model yang digunakan pada tahap ini:

1. *Random Forest*

Random Forest adalah sebuah klasifier yang terdiri dari kumpulan dari klasifirer pohon terstruktur {t(x, Θb), b = 1, ..., B} dimana x adalah input data dan {Θb} adalah sebuah vector acak yang terdistribusi identik dan independen. Setelah pohon terbentuk, setiap pohon memberikan vote untuk kelas paling populer pada masukan x [16].

Random Forest sangat cocok untuk klasifikasi pada data yang cukup besar karena semakin meningkatnya pohon mempengaruhi akurasi yang didapatkan menjadi lebih baik. Keuntungan lain dari *Random Forest,* adalah dapat menangani data yang hilang dan mempertahankan akurasi untuk data yang hilang [16].

Cara kerja *Random Forest* adalah menggunakan data sampel dari dataset latih kedalam pohon keputusan secara acak tetapi dengan penggantian. Setiap pohon selalu bertambah besar dan tidak ada *pruning*. Setelah itu, Setiap pohon yang sudah terbentuk akan melakukan voting berdasarkan label pada data sampel. Hasil *vote* dari setiap label dikombinasikan dan dipilih vote yang memiliki jumlah besar untuk klasifikasi dan rata-rata vote untuk regresi [16].

Dalam melakukan metode klasifikasi, penentuan jumlah simpul pada pohon keputusan bisa melalui *Gini index* atau entropi. Tujuan dari Formula *Gini index* adalah mencari nilai *Gini* dari setiap cabang pada sebuah simpul. Sedangkan Formula Entropi probabilitas hasil tertentu untuk membuat keputusan tentang bagaimana simpul harus bercabang. Rumus *Gini Index*

dapat dilihat pada persamaan Persamaan 2.2 dan Rumus Entropi dapat dilihat pada persamaan Persamaan 2.3 [17].

$$Gini = 1 - \sum_{I=1}^{C}(P_i) \quad (2.2)$$

Persamaan 2.2 Rumus *Gini Index* pada Model Random Forest

$$Entropy = \sum_{I=1}^{C} -(P_i) \times log(P_i) \quad (2.3)$$

Persamaan 2.3 Rumus Entropi pada Model *Random Forest*

Keterangan :

- c = Jumlah kelas
- P_i = Frekuensi relative pada kelas

2. *SVM*

SVM atau *Support Vector Machine* adalah *supervised machine learning* dimana algoritma pelatihan membangun model yang memprediksi label untuk setiap data masukan dengan data contoh yang sudah diberi label dengan menggunakan dua titik vektor [18].

Pada umumnya *SVM* adalah model yang memiliki prinsip klasifikasi linear, tetapi *SVM* bisa dikembangkan agar dapat bekerja pada permasalahan lon-linear dengan menggunakan kernel yang mentransformasikan data ke ruang dimensi yang lebih tinggi atau disebut *kernel trick*. Penggunaan *Kernel trick* memudahkan dalam proses SVM karena untuk menentukan vector support, hanya cukup mengetahu fungsi kernel yang dipakai dan tidak perlu mengetahui wujud fungsi non linear. Secara umum, fungsi kernel yang sering digunakan adalah kernel linear, polynomial dan *Radial basis Function* (RBF) [18].

Tujuan dari algoritma SVM adalah untuk mencari *hyperplane yang terbaik* dalam ruang n-dimensi dengan memisahkan jarak antar kelas menggunakan data yang sudah diberi label atau data latih. *Hyperplane* adalah sebuah garis pemisah dalam membantuk mengklasifikasikan titik data. Untuk memisahkan kedua titik kelas pada data latih, dibutuhkan jenis

hyperplane yang memiliki margin maksimum atau jarak maksimum antara titik data dari kedua kelas [18].

3. *K-Nearest Neighbor*

K-Nearest Neighbor atau disingkat KNN adalah sebuah klasifier yang memberi label terhadap objek berdasarkan jarak dari *k* buah data latih. Nilai *k* menandakan jumlah tertangga terdekat dan syarat nilai *k* harus ganjil dan lebih dari 1 karena kemungkinan tidak ada jawaban saat proses klasifikasi. Penentuan nilai k didasarkan pada jumlah data yang ada dan ukuran dimensi yang dibentuk oleh data tersebut. KNN menentukan ukuran yang sesuai menggunakan persamaan konsinus sebagai penjelasan kedekatan jarak berdasarkan kemiripan dokumen. [19]. Perhitungan kemiripan dokumen dengan metode persamaan kosinus pada model KNN dapat dilihat pada Persamaan 2.4 [19].

$$\cos(\theta_{QD}) = \frac{\sum_{i=1}^{n} Q_i D_i}{\sqrt{\sum_{i=1}^{n}(Q_i)^2} \times \sqrt{\sum_{i=1}^{n}(D_i)^2}} \quad (2.4)$$

Persamaan 2.4 Rumus Persamaan Kosinus

Keterangan :

- $\cos(\theta_{QD})$ = Kemiripan Q terhadap D
- Q = Data Uji
- D = Data latih
- n = Banyaknya data

4. *Multinomial Naïve Bayes*

Multinomial Naïve Bayes adalah sebuah model klassifier yang memberi nilai berdasarkan perhitungan probabilitas jumlah kemunculan pada data latih. Model ini mengasumsikan semua atribut pada data latih independen atau tidak saling ketergantungan yang diberikan oleh nilai pada atribut label. Selain itu, metode ini hanya membutuhkan sedikit data latih dalam proses klasifikasi untuk menentukan estimasi parameter yang diperlukan. [20]. Perhitungan teorema *Bayes* dalam

klasifikasi *Multinomial Naïve Bayes* dapat dilihat pada Persamaan 2.5 [20].

$$P(H|X) = \frac{P(H|X) \times P(H)}{P(X)} \quad (2.5)$$

Persamaan 2.5 Perhitungan teorema *Bayes*

Keterangan :

- X = Data dengan kelas yang belum diketahui
- H = Data Hipotesis pada suatu kelas
- P(H|X) = Probabilitas hipotesiss H bedasarkan kondisi X
- P(H) = Probabilitas hipotesis H
- P(X|H) = Probabilitas X berdasarkan kondisi hipotesis H
- P(X) = Probabilitas hipotesis

2.4 *Confusion Matrix*

Confusion matrix adalah sebuah metode untuk menguji peforma dari model klasifikasi. *Confusion matrix* memberikan informasi actual dan prediksi oleh pengklasifikasi [21]. Tabel yang menggambarkan *Confusion matrix* dapat dilihat pada Tabel 2.1.

	Prediksi Benar	Prediksi Salah
Asli Positif	TP	FN
Asli Negatif	FP	TN

Tabel 2.1 *Confusion Matrix*

Keterangan:

- TP (*True Positive*): Jumlah klasifikasi yang benar dari contoh yang benar
- FN (*False Negative*): Jumlah klasifikasi yang salah dari contoh yang salah
- FP (*False Positive*): Jumlah klasifikasi yang salah dari contoh yang benar
- TN (*True Negative*): Jumlah klasifikasi yang benar dari contoh yang salah

Dari *Confusion matrix,* bisa didapatkan sebuah informasi yaitu *precision, recall* dan *f1-score*. *Precision dan Recall memiliki pengertian yang hampir mirip sehingga susah dibedakan. Precision* adalah rasio prediksi label yang benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positf. *Recall* adalah rasio prediksi label yang benar positif dibandingkan dengan keseluruhan hasil data yang diprediksi label tersebut. Dan *f-1 score* adalah rata-rata dari *precision* dan recall yang dibobotkan. [21]. Pengukuran *precision, recall* dan *f1-score* dapat dilihat pada Persamaan 2.6, Persamaan 2.7 dan Persamaan 2.8 [21].

$$\text{precision} = \frac{\text{TP}}{\text{TP+FN}} \quad (2.6)$$

Persamaan 2.6 Perhitungan *precision*

$$\text{recall} = \frac{\text{TP}}{\text{TP+FP}} \quad (2.7)$$

Persamaan 2.7 Perhitungan *recall*

$$f1score = \frac{2\times\text{recall}\times\text{precision}}{recall+precision} \quad (2.7)$$

Persamaan 2.8 Perhitungan *f1-score*

2.5 Laravel

Pada tugas akhir ini, hasil dataset berita dan COVID-19 akan ditampilkan pada web dengan frameword Laravel, Laravel adalah *framework* aplikasi web dengan sintks yang ekspresif dan elegan. Laravel bertujuan untuk membuat proses pengembangan mudah bagi pengembang tanpa mengorbankan fungsionalitas aplikasi seperti otentikasi, *routing*, *session* dan *caching*. Untuk mencapai tujuannya, Laravel telah mencoba menggabungkan beberapa framework lain yang diimpelementasikan dalam beberapa bahasa seperti Ruby on Rails, ASP.NET MVC dan Sinatra [22]

[*Halaman ini sengaja dikosongkan*]

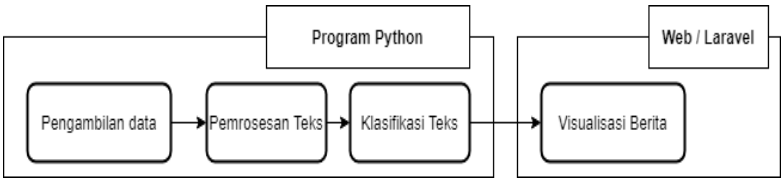
BAB III
ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini membahas tahap analisis dan perancangan desain web visualisasi berita online topik COVID-19 yang akan dibangun. Analisis membahas permasalahan secara umum, desain aplikasi berita online berdasarkan riwayat topik COVID-19, desain model data, desain proses pengumpulan data,desain proses pemrosesan teks , desain proses klasifikasi teks, desain proses visualisasi riwayat berita online.

3.1 Desain Umum Sistem Informasi Berita

Penyelesaian pada tugas akhir ini akan dilakukan menggunakan dua *environment* yaitu python dan web. Program dengan bahasa python akan digunakan untuk pengambilan data, pemrosesan teks dan klasifikasi teks dan dari hasil program tersebut akan berupa data yang nantinya akan ditampilkan melalui web yang akan menggunakan framework Laravel. Diagram alir secara umum akan ditampilkan pada Gambar 3.1.

Gambar 3.1 Desain Umum Sistem Informasi Berita



3.2 Desain Model Data

Langkah awal yang dilakukan adalah merencankan data apa saja yang akan digunakan dalam membuat web berita online topik COVID-19 di Indonesia. Penulis membuat 3 jenis dataset yaitu dataset berita yang diambil dari portal berita online Kompas dan Tribunnews, dataset COVID-19 yang diambil dari spreadsheet kawacovid-19 dan data sampel berita yang akan diambil dari

beberapa dataset berita. Penjelasan setiap dataset terdapat pada Tabel 3.1 dan atribut pada setiap dataset pada Tabel 3.2 dan Tabel 3.3

Tabel 3.1 Penjelasan Dataset Web Riwayat Berita Online COVID-19

Nama Dataset	Keterangan
Berita	Data berita COVID-19 yang diambil dari portal berita Kompas dan Tribunnews
Sampel berita	Data berita COVID-19 yang nantinya akan digunakan untuk melakukan klasifikasi teks
COVID-19	Berbagai macam tabel data yang masing-masing berisi data linimasa COVID-19 pada setiap provinsi di Indonesia berupa jumlah kasus COVID-19, meninggal karena COVID-19 dan sembuh dari COVID-19

Tabel 3.2 Atribut Dataset COVID-19

Atribut	Deskripsi	Contoh
tanggal	Tanggal awal munculnya COVID-19 sampai sekarang	2021-05-23
{nama provinsi}	34 Atribut Provinsi. Setiap atribut memiliki nilai jumlah kasus COVID-19 di provinsi tersebut	234
Total	Jumlah kasus COVID-19 Di seluruh provinsi pada tanggal tersebut	1.775.220

Tabel 3.3 Atribut Dataset Berita dan Sampel Berita

Atribut	**Deskripsi**	**Contoh**
title	Judul berita	UPDATE 31 Maret: Bertambah 1 Orang, Total 3 Pasien Positif Covid-19 asal Gresik
news_portal	Nama Portal Berita	kompas
source	Link URL berita	https://asset.kompas.com/crops/hKqbeF438BgQl5iHgacp57yJHDc=/321x0:3392x2047/750x500/data/photo/2020/03/26/5e7bdcac827b9.jpg
image	Link gambar dari berita	https://asset.kompas.com/crops/hKqbeF438BgQl5iHgacp57yJHDc=/321x0:3392x2047/750x500/data/photo/2020/03/26/5e7bdcac827b9.jpg
date	Tanggal Berita dimuat	2020-03-31
content	Isi dari berita	Satuan Tugas Gugus Pencegahan Penyebaran Covid-19 Kabupaten

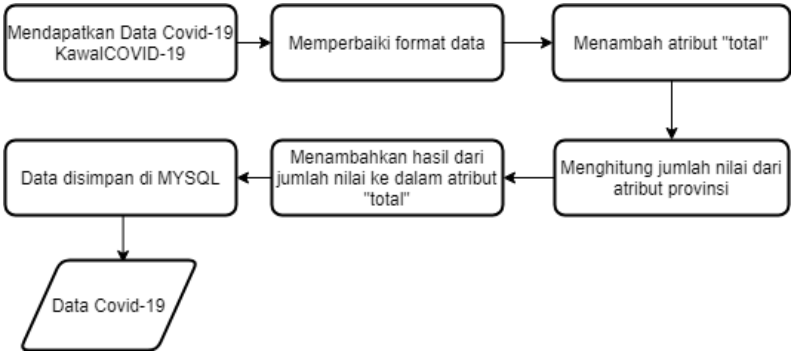
		mengumumkan penambahan satu pasien positif virus corona baru asal Gresik ……
tag	Tag dari berita	Gresik, Virus Corona di Indonesia, Gresik darurat pencegahan Covid-19, gresik positif corona
provinsi	Kategori berita berdasarkan provinsi	jawa timur
kota	Kategori berita berdasarkan kota	gresik
label	Kategori berita berdasarkan label	informasi

3.3 Desain Proses Pengumpulan Data

Pada subbab ini dijelaskan pengumpulan dan pengolahan data COVID-19 dan Berita. Untuk mendapatkan data COVID-19. Penulis memutuskan mengambil data dari spreadsheet yang telah disediakan oleh tim kawalcovid-19. Data yang diambil berupa tanggal dan jumlah COVID-19. Data tersebut nantinya akan diperbarui setiap jam 01.00 WIB dan format data akan diperbaiki sehingga bisa dibaca saat tahap visualisasi COVID-19. Selain itu, penulis menambah informasi baru berupa total kasus COVID-19 di Indonesia yang dihitung dari jumlah kasus provinsi pada setiap

tanggal. Penjelasan tahap pengumpulan dataset COVID-19 akan ditampilkan pada Gambar 3.2.

Gambar 3.2 Diagram Alir Pengumpulan Dataset Covid-19



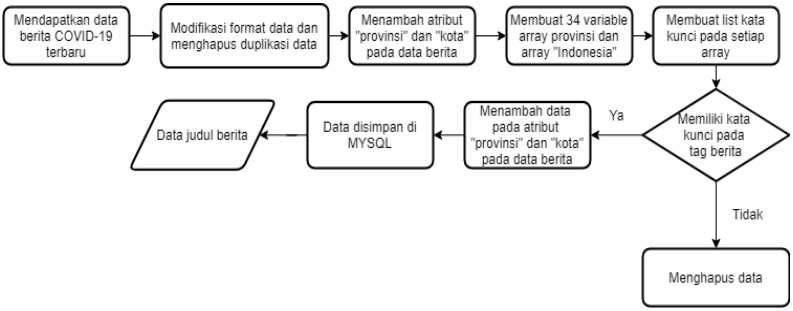
Sedangkan untuk mendapatkan data berita, penulis memutuskan mengambil data dari portal berita online Kompas dan Tribunnews. Portal berita tersebut dipilih selain merupakan protal berita ternama, karena juga menyimpan banyak data berita terkait topik COVID-19 sehingga mudah untuk dilakukan *web scraping*. Proses web scraping yang dilakukan adalah mengambil setiap data artikel pada sebuah kumpulan artikel yang telah dipaginasi atau dibagi menjadi beberapa halaman. Apabila telah mengambil data artikel pada sebuah halaman, maka akan dilakukan pengambilan data di halaman selanjutnya. Proses tersebut akan terus dilakukan sampai halaman terakhir dan akan dimulai lagi dari awal saat jam 12 malam. Setelah mengambil data pada sebuah artikel, data tersebut diubah formatnya karena pada setiap portal berita online memiliki format data masing-masing. Contohnya pada Kompas memiliki format (tahun)/(bulan)/(tanggal). Sedangkan Tribunnews memiliki (tanggal) (bulan) (tahun). Setelah itu artikel yang memiliki data yang sama akan dihapus sehingga tidak ada duplikasi data. Agar bisa mendapatkan data berita terkait COVID-19 di Indonesia terutama pada provinsi di Indonesia, data berita

yang didapatkan akan dibatasi dengan menambahkan atribut provinsi dan kota. Untuk mendapatkan nilai dari atribut provinsi dan juga kota sebuah berita harus memiliki syarat sebagai berikut:

1. Nama Provinsi: Harus memiliki tag yang mengandung kata nama provinsi (panjang/singkatan), Nama Gubernur (panggilan/panjang), nama kota/kabupaten di provinsi tersebut. Apabila memiliki nama kota/kabupaten pada tag tersebut maka nama atribut kota akan ditambahkan kedalam tag tersebut.
2. Indonesia: Harus memiliki tag yang berupa nasional, Indonesia, Presiden RI , Menteri Kesehatan , dan beberapa tag yang berkaitan dengan COVID-19 seperti Vaksin dan PSBB.
3. Apabila sebuah berita bisa masuk kedalam kedua kategori tersebut maka akan di prioritaskan ke bagian provinsi

Dari syarat diatas, sebuah data berita pasti memiliki atribut provinsi jika memiliki atribut kota, tetapi jika memiliki atribut provinsi maka belum tentu memiliki atribut kota. Sementara data berita yang tidak memiliki atribut provinsi akan dihapus. Data yang telah dikumpulkan dimasukkan kedalam dataset berita di MYSQL. Penjelasan singkat tahap pengumpulan dataset Berita COVID-19 akan ditampilkan pada Gambar 3.3.

Gambar 3.3 Diagram Alir Pengumpulan Dataset Berita Covid-19



3.4 Desain Pemrosesan Teks

Selain mendapatkan informasi yang telah disediakan di berita, penulis berinisiatif menambahkan informasi baru yang tidak ada di portal berita online yaitu label. Klasifikasi teks digunakan untuk mendapatkan nilai dari label setiap data berita. Tetapi sebelum itu, penulis menentukan jenis label yang digunakan dan sampel data yang digunakan untuk model pelatihan. Jenis label berita yang digunakan adalah informasi,donasi,kritik,hoaks. Sementara sampel data yang digunakan berasal dari beberapa data pada dataset berita. Atribut data yang diambil untuk dilakukan pemrosesan teks adalah judul berita. Data judul berita diambil karena judul berita menggambarkan garis besar dari berita. Dalam memilih data berita yang akan dimasukkan kedalam sampel data berita, penulis membuat kata kunci untuk setiap label berita. Apabila judul berita mengandung kata kunci tersebut, maka diberi nama label sesuai dengan kategori kata kunci dan akan dimasukkan ke sampel berita. Penjelasan jenis label berita yang digunakan pada tugas akhir ini dapat dilihat pada Tabel 3.4.

Tabel 3.4 Jenis Label

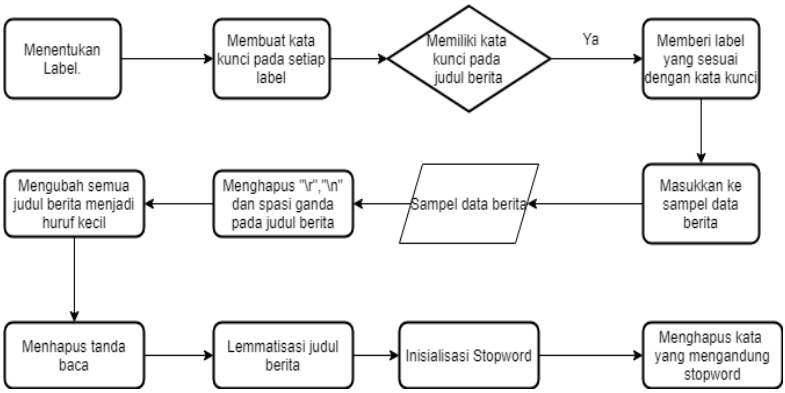
Nama Label	**Keterangan**	**Contoh Judul Berita**	**Kata Kunci**
Informasi	Pengumuman jumlah peningkatan atau penurunan kasus COVID-19	UPDATE 9 April: Tambah 121, Total 42.348 Orang Meninggal Dunia akibat Covid-19 di Indonesia	update, kasus, tingkat, sembuh, meninggal
Donasi	Donasi atau bantuan dalam upaya	12 Ton Peralatan Kesehatan Yang	donasi, bantu, apd, dana, salur

	pencegahan dan pengendalian wabah	Baru Tiba di Indonesia Bantuan Pemerintah Tiongkok	
Kritik	Untuk mempertanyakan tindakan pemerintah pusat atau daerah atau pihak lain terhadap kebijakan atau tindakan terkait COVID-19	4 Kritik Anggota Komisi IX DPR Agar Pemerintah Fokus Tangani Corona	Kritik, , kebijakan, tuduh, minta, buruk
Hoaks	Klarifikasi Berita Palsu, Menanggapi rumor atau kritikan	12 Karyawan Pertamina ODP Covid-19 Diusir dari Karantina, Dianggap Tak Patuh, Berikut Faktanya!	hoaks, klaim, klarifikasi, tanggap,fakta

Judul berita pada dataset sampel berita tersebut akan dilakukan pemrosesan teks sebelum masuk tahap klasifikasi teks . Tahap pertama dalam pemrosesan teks adalah menghapus tanda baca pada setiap data judul berita. Kedua adalah menghapus tab, enter atau spasi ganda pada judul berita. Ketiga adalah letimasi teks atau proses mengembalikan kata-kata infleksi menjadi kata dasar. Yang terakhir adalah menghapus kata-kata yang termasuk dalam list *stopword*. List *stopword* dibuat berdasarkan Natural Language Toolkit (NLTK) pada bahasa Indonesia. Penulis juga menambahkan beberapa kata seperti “covid-19”, “korona”,

“Indonesia”, dan beberapa kata lainnya agar meningkatkan kualitas fitur dalam proses pelatihan model . Penjelasan singkat tahap pemrosesan teks dapat dilihat pada Gambar 3.4.

Gambar 3.4 Diagram Alir Pemrosesan Teks Untuk Proses Klasifikasi Teks



3.5 Desain Klasifikasi Teks

Pada subbab ini akan dijelaskan proses klasifikasi teks pada judul berita di dataset sampel berita. Subbab ini akan dibagi menjadi 2 bagian yaitu rekayasa fitur dan pemberian label

3.5.1 Rekayasa Fitur

Rekayasa fitur adalah proses mengubah data menjadi fitur untuk bertindak sebagai model pembelajaran mesin dalam meningkatkan kinerja model. Pada tahap ini telah dilakukan pemrosesan teks sesuai pada Gambar 3.4.

Setiap data sampel berita nantinya akan diberikan kode sesuai nilai dari atribut label. Pelabelan kode bertujuan untuk memudahkan proses model pembelajaran mesin yang membutuhkan fitur numerik dan nama label dalam memberikan nilai label.

Penulis membutuhkan sebuah set data latih untuk membuat model dan data tes untuk membuktikkan kualitas model ketika mendapatkan data atribut label yang tak terlihat. Penulis membagi 85% data latih dan 15% data tes. Keluaran dari hasil ini adalah mendapatkan nilai x_test, y_train, x_train dan y_train.

Tahap ketiga dalam rekayasa fitur adalah teks representasi. penulis pada tahap ini menggunakan TF-IDF vector sebagai fitur untuk memberikan bobot pada setiap kata pada dokumen di dalam *corpus*. Keluaran dari tahap ini adalah fitur test, label test, fitur train dan label train.

3.5.2 Latih Model

Output yang dikeluarkan pada tahap rekayasa fitur akan digunakan dalam proses latih model. Pengujian setiap jenis model dilakukan untuk mendapaktan model terbaik untuk pelabelan atribut label.

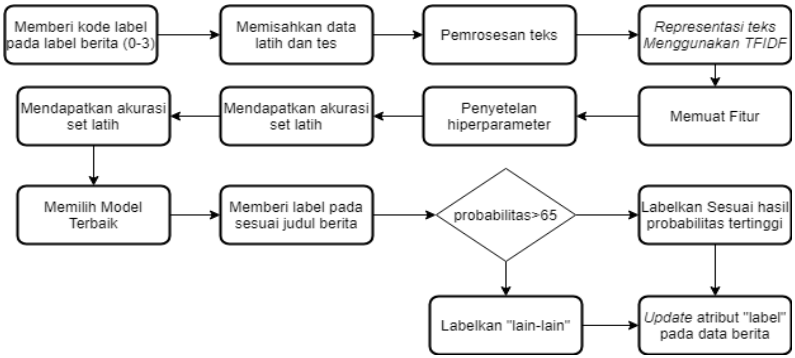
Pada setiap jenis model akan dilakukan *randomized search cross validation* untuk mendapatkan hyperparameter dengan akurasi terbaik. Proses *randomized search cross validation* menggunakan 3-Fold Cross Validation dengan 50 iterasi untuk memperpendek waktu eksekusi. Kemudian penulis melakukan *grid search cross validation* dengan dari hasil hyperparameter terbaik dari hasil *randomized search cross validation*. Tahap ini dilakukan bertujuan agar mengurangi waktu ekseskusi, tetapi dapat mencakup nilai hiperparameter yang sangat luas dan mendapat akurasi yang tinggi.

Setiap model akan dilakukan pelatihan model untuk mendapatkan akurasi data latih dan Hasil dari pengujian performa dibandingkan pada setiap model. Model dengan akurasi akurasi set latih terbaik akan dipilih menjadi model yang akan digunakan untuk memberikan atribut label pada setiap data berita.

Sebelum melakukan pelabelan berita, penulis menentukan ambang batas atau probabilitas bersyarat untuk menentukan atribut label sebesar 65%. Apabila data berita memiliki salah satu nama label sesuai Tabel 3.4 dengan probabilitas bersyarat diatas 65%

maka atribut label akan diberi nilai sesuai nama atribut tersebut. Jika ada dua nama label yang memiliki probabilitas bersyarat diatas 65%, maka akan dipilih nama label yang memiliki probablitas tertinggi. Data berita yang tidak memiliki nama label yang diatas 65% akan diberi nilai "lain-lain". Penjelasan tahap klasifikasi teks dapat dilihat pada Gambar 3.5

Gambar 3.5 Diagram Alir Klasifikasi Teks



3.6 Desain Visualisasi Web

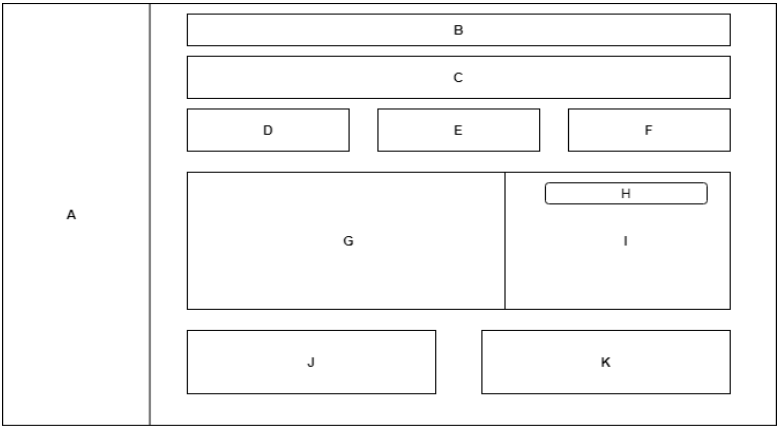
Pada subbab ini adalah tahap visualisasi hasil data yang telah didapatkan dari *web scraping* dan diolah ke dalam sebuah antarmuka web. Antarmuka web yang berhubungan lansung dengan pengguna harus memiliki tampilan yang rapi dan menarik bagi pengguna dan kemudahan pengguna dalam memahami maksud dari fitur visualisasi yang ada.

Sistem memiliki 4 antarmuka web yaitu halaman utama Riwayat Berita COVID-19 , halaman Daftar Riwayat Berita COVID-19, halaman Pencarian COVID-19, dan halaman Statistik Berita COVID-19.

3.6.1 Desain Halaman Utama Riwayat Berita COVID-19

Halaman ini digunakan untuk menampilkan data covid-19 dan data berita terbaru. Data Covid-19 yang ditampilkan berupa jumlah kasus, meninggal , dan sembuh dalam bentuk angka dan *bar chart*. Sedangkan data berita yang ditampilkan adalah riwayat berita terbaru tetapi hanya dibatasi sebanyak 100 berita. Desain halaman utama berita COVID-19 ditampilkan pada Gambar 3.6 dan penjelasan desain halaman Utama dijelaskan pada Tabel 3.5.

Gambar 3.6 Desain Halaman Utama Riwayat Berita COVID-19



Tabel 3.5 Penjelasan Desain Halaman Utama Riwayat Berita COVID-19

Label Atri-but	**Nama Atribut**	**Jenis Atribut**	**Kegunaan**	**Jenis Masukan/ Keluaran**
A	Menu	Tombol	Berisi kumpulan tombol yang mengarah ke	*ButtonClick*

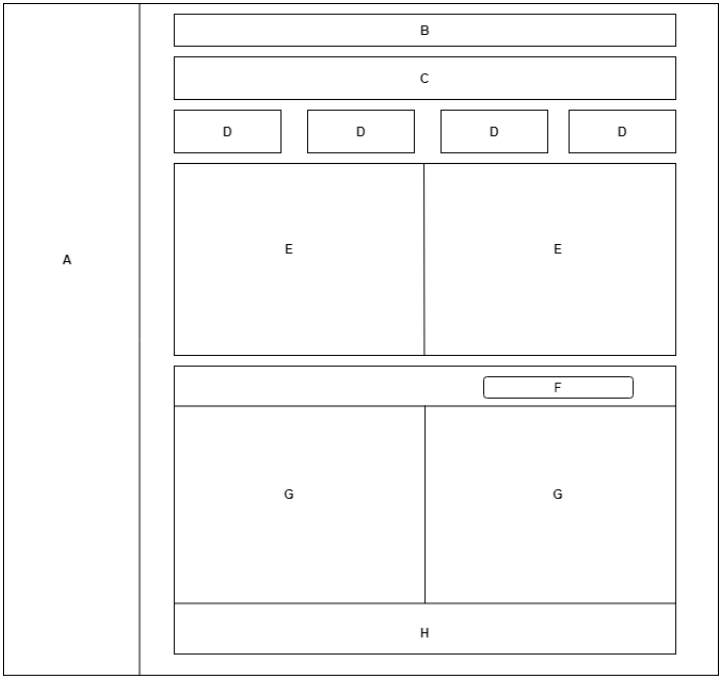
			halaman lainnya	
B	Judul Halaman	*String*	Menampilkan Judul Halaman	*String*
C	Filter Data	Formulir	Menyaring data covid-19 dan data berita	2 input tanggal, 1 *dropdown* dan 1 tombol *submit*
D	Data kasus COVID-19	*String*	Jumlah kasus COVID-19	Angka
E	Data kasus meninggal COVID-19	*String*	Jumlah kasus meninggal karena COVID-19	Angka
F	Data kasus sembuh COVID-19	*String*	Jumlah kasus sembuh dari COVID-19	Angka
G	Data Covid-19	Grafik	Graf kasus, meninggal dan sembuh COVID-19	*Line Chart*
H	Filter riwayat berita	Formulir	Menyaring data berita berdasarkan judul	*String*
I	Data Berita	Gambar dan *string*	Menampilkan 100 data berita terbaru	Gambar dan *string*
J	Berita	Tombol	Menuju halaman daftar berita	*ButtonClick*

K	Statistik	Tombol	Menuju halaman statistic berita	*ButtonClick*

3.6.2 Desain Halaman Daftar Riwayat Berita COVID-19

Halaman ini digunakan untuk menampilkan daftar riwayat berita COVID-19 di seluruh Indonesia dan provinsi/kota data berita terbaru. Selain itu apabila ingin menyaring data berita untuk mendapatkan riwayat berita COVID-19 di setiap provinsi, halaman juga akan menampilkan daftar kota pada provinsi tersebut. Desain halaman daftar riwayat berita COVID-19 ditampilkan pada Gambar 3.7 dan penjelasan desain halaman daftar riwayat berita dijelaskan pada Tabel 3.6.

Gambar 3.7 Desain Halaman Daftar Riwayat Berita COVID-19



Tabel 3.6 Penjelasan Desain Halaman Daftar Riwayat Berita COVID-19

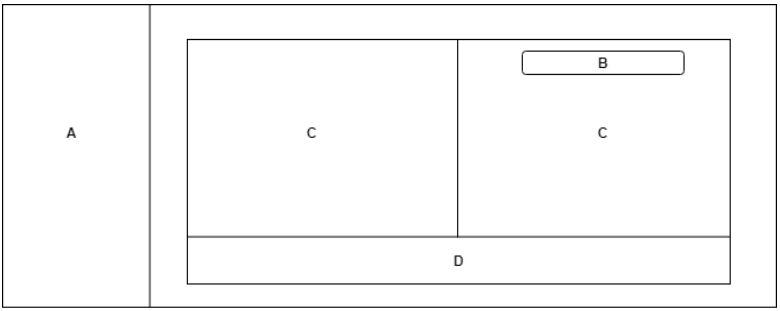
Label Atri-but	**Nama Atribut**	**Jenis Atribut**	**Kegunaan**	**Jenis Masukan/ Keluaran**
A	Menu	Tombol	Berisi kumpulan tombol yang mengarah ke halaman lainnya	*ButtonClick*
B	Judul Halaman	*String*	Menampilkan Judul Halaman	*String*
C	Filter Berita	Formulir	Menyaring data berita	2 input tanggal, 2 *dropdown* dan 1 tombol *submit*
D	List Kota	Tombol	Menuju halaman daftar berita berdasarkan nama kota pada tombol tersebut	*ButtonClick*
E	Riwayat Berita Terbaru	Gambar dan *string*	Menampilkan 4 data berita terbaru	Gambar dan *string*
F	Filter Judul Berita	Formulir	Menyaring data berita berdasarkan judul	*String*

G	Riwayat Berita	Gambar dan *string*	Menampilkan 10 data berita	Gambar dan *string*
H	*Paginatio n* riwayat berita	*Paginati-on*	Membagi setiap 10 data berita ke halaman daftar berita yang baru	*ButtonClick*

3.6.3 Desain Halaman Pencarian Berita COVID-19

Halaman ini digunakan untuk menampilkan daftar riwayat berita COVID-19 berdasarkan hasil pencarian judul berita pada halaman daftar riwayat berita. Desain halaman cari judul berita COVID-19 ditampilkan pada Gambar 3.8 dan penjelasan desain halaman cari judul berita dijelaskan pada Tabel 3.7.

Gambar 3.8 Desain Halaman Cari Judul Berita COVID-19



Tabel 3.7 Penjelasan Desain Halaman Cari Judul Berita COVID-19

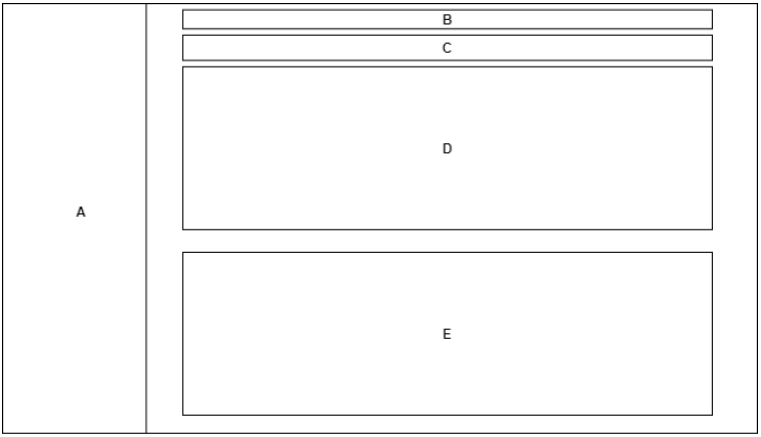
Label Atri-but	**Nama Atribut**	**Jenis Atribut**	**Kegunaan**	**Jenis Masukan/ Keluaran**
A	Menu	Tombol	Berisi kumpulan	*ButtonClick*

			tombol yang mengarah ke halaman lainnya	
B	Filter Judul Berita	Formulir	Menyaring data berita berdasarkan judul	*String*
C	Riwayat Berita	Gambar dan *string*	Menampilkan 10 data berita berdasarkan hasil pencarian judul berita	Gambar dan *string*
D	*Paginati-on* riwayat berita	*Paginati-on*	Membagi setiap 10 data berita ke halaman daftar berita yang baru	*ButtonClick*

3.6.4 Desain Halaman Statistik Riwayat Berita COVID-19

Halaman ini digunakan untuk menampilkan statistik riwayat berita COVID-19 berdasarkan label berita. Tampilan desain menggunakan *bar chart* dan *stacked bar chart* untuk menampilkan jumlah berita menggunakan label sesuai pada Tabel 3.4. Desain halaman cari judul berita COVID-19 ditampilkan pada Gambar 3.9 dan penjelasan desain halaman cari judul berita dijelaskan pada Tabel 3.8.

Gambar 3.9 Desain Halaman Statistik Riwayat Berita COVID-19



Tabel 3.8 Penjelasan Desain Halaman Statistik Riwayat Berita COVID-19

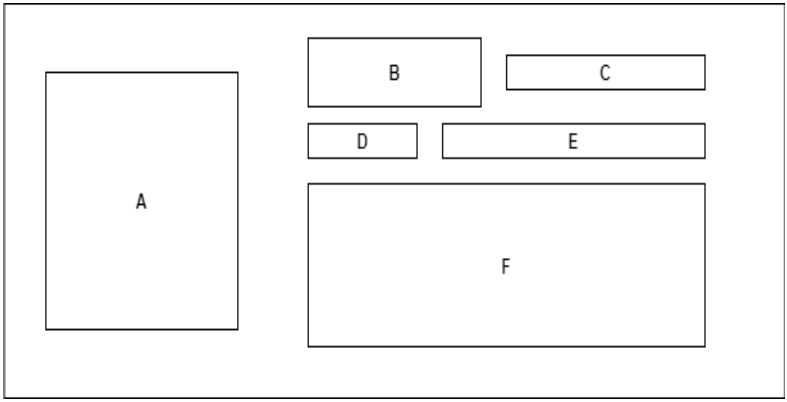
Label Atrib-ut	**Nama Atribut**	**Jenis Atribut**	**Kegunaan**	**Jenis Masukan/ Keluaran**
A	Menu	Tombol	Berisi kumpulan tombol yang mengarah ke halaman lainnya	*ButtonClick*
B	Judul Halaman	*String*	Menampilkan Judul Halaman	*String*
C	Filter Data	Formulir	Menyaring data	2 input tanggal, 1 *dropdown* dan 1 tombol *submit*

C	*Bar chart*	Grafik	Menampilkan *Bar chart*	*G*rafik
D	*Stacked bar chart*	Grafik	Menampilkan *Stacked bar chart*	Grafik

3.6.5 Desain Tampilan Data Berita

Subbab ini menjelaskan tampilan data berita yang akan selalu ditampilkan pada web sistem informasi riwayat berita. Desain tampilan data berita ditampilkan pada Gambar 3.10 dan penjelasan data berita dijelaskan pada Tabel 3.8.

Gambar 3.10 Desain Tampilan Data Berita COVID-19



Tabel 3.9 PenjelasanDesain Tampilan Data Berita

Label Atrib-ut	**Nama Atribut**	**Jenis Atribut**	**Kegunaan**	**Jenis Masukan/ Keluaran**
A	Gambar	Gambar	Menampilkan Gambar Berita	Gambar

B	Portal Berita	*String*	Menampilkan Nama portal berita online	*String*
C	Tanggal	Tanggal	Menampilkan halaman berita pada tanggal tersebut	ButtonClick
D	Label	*String*	Menampilkan halaman berita pada label tersebut	*String*
E	Tempat	*String*	Menampilkan halaman berita pada tempat tersebut	*String*
F	Judul	*String*	Menampilkan halaman asli berita itu dimuat	*String*

[*Halaman ini sengaja dikosongkan*]

BAB IV
IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi sistem pada tiap tahap berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Implementasi berupa langkah-langkah dan kode sumber untuk membangun sebuah sistem pada setiap tahapannya.

4.1 Lingkungan Implementasi

Pada tugas akhir ini, digunakan beberapa perangkat yang berguna untuk mempermudah pengerjaan tugas akhir. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dapat dilihat pada Tabel 4.1.

Tabel 4.1 Spesifikasi Lingkungan Implementasi

No.	Jenis Perangkat	Spesifikasi
1	Perangkat Keras	• *Processor*: Intel(R) Core(TM) i7-7200U CPU @ 2.50GHz • *Random Access Memory*: 8GB
2	Perangkat Lunak	• *Operating System*: Windows 10 Home Single Language • Bahasa pemrograman Python 3.8.5 64-bit • *Integrated Development Environment*: Jupyter Notebook • *Text Editor* Sublime Text 3

4.2 Implementasi Pengumpulan data

Subbab ini menjelaskan implementasi program pengumpulan data COVID-19 dan berita yang akan ditampilkan pada sistem web. Beberapa data berita juga digunakan sebagai data sampel berita yang digunakan sebagai data latih dalam proses

klasifikasi teks. Program pengumpulan data ditulis menggunakan bahasa python.

4.2.1 Data COVID-19

Program penulis membagi menjadi beberapa program untuk mengambil Data kasus,sembuh dan meninggal COVID-19. Meskipun tiap program memiliki tujuan berbeda tetapi memiliki kesamaan dalam isi program dan hanya berbeda dalam pengambilan baris awal dan akhir dan tujuan *update* tabel data. Oleh karena itu, pada subbab ini akan dijelaskan pengumpulan data kasus COVID-19. Data COVID-19 menggunakan pustaka yang ditunjukkan pada Kode Sumber 4.1. Penjelasan fungsi pustaka yang digunakan terdapat pada Tabel 4.2.

```
1. import mysql.connector
2. import gspread
3. from oauth2client.service_account import
   ServiceAccountCredentials
4. import datetime
5. import time
6. from datetime import timedelta
```

Kode Sumber 4.1 Inisialisasi *Library* Program Pengumpulan Data COVID-19

Tabel 4.2 Nama dan Penjelasan Pustaka dalam Pengumpulan Data COVID-19

No.	Nama Library	Penjelasan
1	gspread	Pustaka API python untuk google Sheets
2	oauth2client.service_account	Pustaka python untuk mendapatkan otorisasi API dari Kredensial Google API
3	time	Pustaka python untuk menangani tugas yang berhubungan dengan

		waktu. Pada program ini berfungsi untuk menangguhkan waktu eksekusi menggunakan fitur *sleep()*
4	datetime	Pustaka python yang menggunakan tanggal sebagai objek tanggal
5	mysql.connector	Pustaka python untuk memperbolehkan program python terhubung dengan database MySQL

Pada implementasi data COVID-19 tahap pertama yang akan dilakukan adalah menghubungkan ke *database* MYSQL server sesuai dengan username dan password. Karena data diupdate setiap jam 01.00 W, penulis mengambil tanggal terakhir data covid-19 diambil yang dijadikan patokan awal untuk mengambil data covid-19 setelah tanggal tersebut. Implementasi program pada tahap ini dapat dilihat pada Kode Sumber 4.2.

```
1. mydb = mysql.connector.connect(
2.   host="localhost",
3.   user="pmauser",
4.   password="password_here",
5.   database="tacovid"
6. )
7.
8. mycursor = mydb.cursor()
9. mycursor.execute("SELECT   tanggal   FROM
   data_covid19_kasus ORDER BY tanggal DESC
   limit 1")
10. date = mycursor.fetchall()
```

Kode Sumber 4.2 Inisialisasi Pengumpulan Data Kasus COVID-19

Setelah itu, program akan mendapatkan otorisasi dari Google API untuk mendapatkan akses mendapatkan data spreadsheet Data Covid-19 kawalcovid19 dari Google API. Dikarenakan tabel kasus, sembuh dan meninggal COVID-19 menjadi satu *sheet*, setiap program data COVID-19 perlu menentukan baris data terbaru dan terakhir. Program akan mengambil data spreadsheet pada baris diantara baris data terbaru. Data tersebut nantinya dimasukkan ke dalam database MYSQL. Implementasi program pada tahap ini dapat dilihat pada Kode Sumber 4.3.

```
1.  #mendapatkan izin pengambilan data menggunakan
    Google API
2.  scope =
    ["https://spreadsheets.google.com/feeds",'https://
    www.googleapis.com/auth/spreadsheets',"https://www
    .googleapis.com/auth/drive.file","https://www.goog
    leapis.com/auth/drive"]
3.  creds =
    ServiceAccountCredentials.from_json_keyfile_name("
    creds.json", scope)
4.  client = gspread.authorize(creds)
5.
6.  #membuka link spreadsheet
7.  sheet =
    client.open_by_url("https://docs.google.com/spread
    sheets/d/1ma1T9hWbec1pXlwZ89WakRk-
    OfVUQZsOCFl4FwZxzVw/edit#gid=2052139453").workshee
    t("Timeline")  # Open the spreadhseet
8.
9.  #mendapatkan row awal dan akhir
10. first_date = datetime.date(2020, 3, 18)
11. last_date= tanggal[0][0]+ timedelta(days=1)
12. today_date = datetime.date.today()
13. column_row=sheet.find("Total Kasus")
14. first_row= last_date-first_date
15. last_row = today_date - first_date
16.
17. for day in
    range(first_row.days+column_row+1,last_row.days+co
    lumn_row):
18.     row = sheet.row_values(day)
19.     print(row)
20.
21.     sql = """INSERT INTO data_covid19_kasus
    (tanggal,aceh,bali,banten,babel,bengkulu,diy,
```

```
        jakarta,jambi,jabar,jateng,jatim,kalbar,kaltim,Kal
        teng,Kalsel,Kaltara,kep_riau,NTB,Sumsel,Sumbar,Sul
        ut,Sumut,Sultra,Sulsel,Sulteng,Lampung,Riau,Malut,
        Maluku,Papbar,Papua,Sulbar,NTT,Gorontalo) VALUES
        (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)"""
    22.     val = (last_date,row[1].replace(",",
        ""),row[2].replace(",", ""),row[3].replace(",",
        ""),row[4].replace(",", ""),row[5].replace(",",
        ""),row[6].replace(",", ""),row[7].replace(",",
        ""),row[8].replace(",", ""),row[9].replace(",",
        ""),row[10].replace(",", ""),row[11].replace(",",
        ""),row[12].replace(",", ""),row[13].replace(",",
        ""),row[14].replace(",", ""),row[15].replace(",",
        ""),row[16].replace(",", ""),row[17].replace(",",
        ""),row[18].replace(",", ""),row[19].replace(",",
        ""),row[20].replace(",", ""),row[21].replace(",",
        ""),row[22].replace(",", ""),row[23].replace(",",
        ""),row[24].replace(",", ""),row[25].replace(",",
        ""),row[26].replace(",", ""),row[27].replace(",",
        ""),row[28].replace(",", ""),row[29].replace(",",
        ""),row[30].replace(",", ""),row[31].replace(",",
        ""),row[32].replace(",", ""),row[33].replace(",",
        ""),row[34].replace(",", ""))
    23.     mycursor.execute(sql, val)
    24.     mydb.commit()
    25.     last_date=last_date + timedelta(days=1)
    26.     time.sleep(0.5)
    27.
    28. mycursor.close() mycursor.close()
```

Kode Sumber 4.3 Pengumpulan Program Pengumpulan Data Kasus COVID-19

Semua data COVID-19 di seluruh provinsi Indonesia telah didapatkan, tetapi tidak ada data total kasus covid-19 di Indonesia pada setiap tanggal. Sehingga penulis menambah informasi baru pada data COVID-19 dengan menambahkan atribut total berupa jumlah kasus data COVID-19 di seluruh provinsi Indonesia. Setiap data COVID-19 yang memiliki nilai atribut total 0, dilakukan perubahan nilai nengan menjumlahkan nilai atribut seluruh

provinsi di Indonesia. Implementasi program pada tahap ini dapat dilihat pada Kode Sumber 4.4.

```
1. mycursor = mydb.cursor()
2. mycursor.execute("SELECT * FROM
   data_covid19_kasus where total=0")
3. myresult = mycursor.fetchall()
4. for row in myresult:
5.     total=0
6.     year=row[0].strftime("%Y")
7.     month=row[0].strftime("%m")
8.     day=row[0].strftime("%d")
9.     date=year+"-"+month+"-"+day
10.    print(date)
11.    for i in range(1,35):
12.        total=total+row[i]
13.    sql = "UPDATE data_covid19_kasus SET
   total ='"+str(total)+"' WHERE
   tanggal='"+date+"'"
14.    mycursor.execute(sql)
15.    mydb.commit()
16.          print(total)
```

Kode Sumber 4.4 Potongan Kode Fungsi Penghubung Ke Fungsi Kombinasi Himpunan

4.2.1 Data Berita

Pengambilan data berita pada portal berita online mengunakan *web scraping* dengan framework Scrapy dan bahasa Python.Pengambilan data dilakukan pada setiap 1 jam. Penjelasan fungsi pustaka yang digunakan pada Data Berita terdapat pada Tabel 4.3.

Tabel 4.3 Nama dan Penjelasan Pustaka dalam Pengumpulan Data Berita

Nama Library	**Penjelasan**
scrapy	Pustaka python untuk mengekstrak data yang dibutuhkan dari situs web

datetime	Pustaka python yang menggunakan tanggal sebagai objek tanggal
mysql.connector	Pustaka python untuk memperbolehkan program python terhubung dengan database MySQL

Berikut langkah-langkah proses scraping berita pada Berita Online:

1. Menentukan link portal berita online yang dituju. Penulis memutuskan link yang dituju berupa link portal berita yang berisikan semua berita yang memiliki topic COVID-19. Proses ini dapat dilihat pada Kode Sumber 4.4.

```
1.  import scrapy
2.  import datetime
3.  from tribunnews.items import TribunnewsItem
4.
5.  class Covid19Spider(scrapy.Spider):
6.      name = 'covid-19'
7.      allowed_domains = ['tribunnews.com']
8.      start_urls =
    ['https://www.tribunnews.com/tag/covid-19?page=1']
```

Kode Sumber 4.4 Menentukan tujuan *scraping* portal berita

2. Mendapatkan link halaman artikel berita pada setiap halaman daftar berita dengan mengambil elemen HTML untuk menuju link tersebut. Elemen HTML didapatkan dengan melakukan inspeksi web. Kode sumber pada proses ini dapat dilihat pada Kode Sumber 4.5.

```
1.  def parse(self, response):
2.    pages = response.xpath('//*[@class="article__list
    clearfix"]')
3.  #menuju artikel berita
4.  for page in pages:
5.      links =
    page.xpath('.//*[@class="article__link"]/@href').ex
    tract_first()+"?page=all"
6.      absolute_next_url = response.urljoin(links)
7.      yield scrapy.Request(absolute_next_url,
    callback=self.parse_artikel,  )
```

```
8.      cek_len= len(start_urls)-len(response.url)
9.
10.     if(int(response.url[cek_len:])+1<=100):
11.       next_page_url = response.url[0:cek_len] +
    str(int(response.url[cek_len:]) +1)
12.       absolute_next_page_url=
    response.urljoin(next_page_url)
13.       yield scrapy.Request(absolute_next_page_url)
14.
15.     else:
          sys.exit()
```

Kode Sumber 4.5 Mendapatkan daftar artikel berita

3. Setiap artikel berita diambil judul berita, tanggal berita, isi berita, tag berita , link berita dan link gambar berita dengan menggunakan elemen HTML untuk mendapatkan data tersebut. Kode sumber pada proses ini dapat dilihat pada Kode Sumber 4.6.

```
1.  def parse_page(self, response):
2.    items = KompasItem()
3.    judul = response.xpath(
    '//*[@class="read__title"]/text()').extract_first()
4.    waktu =
    response.xpath('//*[@class="read__time"]/text()').e
    xtract_first().split()[2][:-1]
5.     artikel =
    response.xpath('//*[@class="read__content"]')
6.     konten_array=
    artikel.xpath('//*/p/text()').extract()[1:-6]
7.     tag_array =
    response.xpath('//*[@class="tag__article__item"]/a/
    text()').extract()
8.     link = response.url
9.      gambar_link_array=
    response.xpath('//*/div/img/@src').extract()
```

Kode Sumber 4.6 Mendapatkan link artikel berita

4. Melakukan perubahan format ke variabel konten_array, tag_array dan gambar_link_array dari array menjadi string. Variabel waktu juga dirubah formatnya menjadi waktu tanggal

sesuai format MYSQL. Kode sumber pada proses ini dapat dilihat pada Kode Sumber 4.7.

```
tahun=waktu[6:10]
bulan=waktu[3:5]
hari=waktu[0:2]
tanggal=tahun+"-"+bulan+"-"+hari
gambar_link=""
  for img in gambar_link_array:
     if("/crops/" in img):
       gambar_link= img
       break;

konten=""
  for i in range(len(konten_array)):
    for j in konten_array[i].split():
      konten += j+' '

tag=""
  for i in range(len(tag_array)):
    if(i!=len(tag_array)-1):
       tag += tag_array[i]+', '
    else:
       tag += tag_array[i]
```

Kode Sumber 4.7 Merubah Format Data

5. Menambah atribut provinsi dan kota sesuai kata kunci pada tag berita. Penulis membuat variable bernama provinsi yang berisikan tentang 34 variable list nama provinsi di Indonesia. Variable nama provinsi tersebut berisikan kata kunci yang dapat dilihat pada Lampiran. Selain itu penulis juga membuat variable nasional yang berisikan beberapa kata kunci. Kode sumber pada proses penambahan atribut kota dan provinsi dapat dilihat pada Kode Sumber 4.8.

```
for row in data_berita:
  tags_word=''
  area=""
  kota=""
  cek=0
```

```
for province in provinsi:
      for tag_area in province:
        tag_berita = row[1].lower()
        cari = r"\b"+ tag_area +r"\b"
        hasil = re.search(cari, tag_berita)
        if str(hasil)!="None":
          if(tag_area== "maluku" and "maluku utara"
in row[1].lower()):
            break
          elif(tag_area== "papua" and "papua barat"
in row[1].lower()):
            break
    elif(tag_area== "riau" and "kepulauan riau" in
row[1].lower()):
             break
           else:
             if(cek==1):
               area+=", "
             elif(kota==""):
               for i in range(3,len(province)):
                 tag_berita1 = row[0].lower()
                 tag_berita2 = row[1].lower()
                 cari = r"\b"+ province[i] +r"\b"
                 hasil1 = re.search(cari,
tag_berita1)
                 hasil2 = re.search(cari,
tag_berita2)
                 if(str(hasil1)!="None" or
str(hasil2)!="None"):
                   kota=province[i]
                   break
  if(cek==0):
    for tag_area in nasional:
      tag_berita = row[1].lower()
      cari = r"\b"+ tag_area +r"\b"
      hasil = re.search(cari, tag_berita)
      if str(hasil)!="None":
        area+= "indonesia"
        break
```

Kode Sumber 4.8 Menambah Atribut Berita

6. Setelah mendapatkan atribut berita. Data Berita disimpan di mysql. Data yang tidak memiliki nilai atribut provinsi dihapus.

4.3 Implementasi Klasifikasi Teks

Subbab ini menjelaskan implementasi klasifikasi teks untuk mendapatkan label atribut data pada data berita. Program ini ditulis menggunakan bahasa pemrograman Python.

4.3.1 Penggunaan Pustaka

Penjelasan dan fungsi dari pustaka yang digunakan selama proses klasifikasi teks terdapat pada Tabel 4.3.

Nama Library	Penjelasan
pickle	Pustaka yang mengimplementasikan protokol biner untuk serialisasi dan de-serialisasi struktur objek Python
pandas	Pustaka yang menyediakan struktur data dan analisis data yang mudah digunakan
nltk	Pustaka untuk *natural language processing*. Pada tugas akhir pustaka ini digunakan untuk mendapatkan list *stopword* dalam bahasa indonesia
sklearn	Bisa disebut pustaka sckit-learn. Pustaka ini adalah pustaka untuk membantu pembelajaran mesin dalam python
Sastrawi.Stemmer .StemmerFactory	Pustaka untuk mengembalikan kata infleksi dalam Bahasa Indonesia menjadi kata dasar
Scikit-sklearn	Pustaka untuk pembelajaran mesin. Penggunaan pustaka ini mudah dan efisien untuk analisis data prediktif
numpy	Pustaka untuk kompustasi ilmiah yang berhubungan dengan *array* dan *matrix*
matplotlib.pyplot	Pustaka untuk membuat visualisasi statis, animasi, dan interaktif

4.3.2 Implementasi Pemrosesan Teks

Pada tahap ini setiap judul berita dilakukan pemrosesan teks menggunakan program dengan bahasa pemrograman Python. Hasil dari pemrosesan teks digunakan dalam pelatihan model. Berikut beberapa tahap implementasi program pemrosesan teks. Implementasi tahap pemrosesan teks dapat dilihat pada Kode Sumber 4.9, Kode Sumber 4.10, Kode Sumber 4.11, Kode Sumber 4.12, Kode Sumber 4.13, dan Kode Sumber 4.14

a. Inisialisasi pustaka

```
1.  import pickle
2.  import pandas as pd
3.  import re
4.  import nltk
5.  from nltk.corpus import stopwords
6.  from Sastrawi.Stemmer.StemmerFactory import
    StemmerFactory
7.  from sklearn.feature_extraction.text import
    TfidfVectorizer
8.  from sklearn.model_selection import
    train_test_split
9.  from sklearn.feature_selection import chi2
10. import numpy as np
11. import matplotlib.pyplot as plt
```

Kode Sumber 4.9 Inisialisasi Pustaka

b. Menghilangkan spasi atau tab

```
1.  df['title_parsed_1'] =
    df['title'].str.replace("\r", " ")
2.  df['title_parsed_1'] =
    df['title_parsed_1'].str.replace("\n", " ")
3.  df['title_parsed_1'] =
    df['title_parsed_1'].str.replace("    ", " ")
```

Kode Sumber 4.10 Menghilangkan spasi atau tab

c. Mengubah menjadi huruf kecil

```
1.  df['title_parsed_2'] =
    df['title_parsed_1'].str.lower()
```

Kode Sumber 4.11 Mengubah menjadi huruf kecil

d. Menghilangkan tanda baca dan angka

```
punctuation_signs = list("!\"#$%&()*+-
./:;<=>?@[\]^_`{|}~\0123456789")
df['title_parsed_3'] = df['title_parsed_2']

for punct_sign in punctuation_signs:
    df['title_parsed_3'] =
df['title_parsed_3'].str.replace(punct_sign, '')
```

Kode Sumber 4.12 Mendapatkan link artikel berita

e. Lemmatisasi teks

```
factory = StemmerFactory()
stemmer = factory.create_stemmer()
nrows = len(df)
lemmatized_text_list = []

for row in range(0, nrows):

    lemmatized_list = []
    text = df.loc[row]['title_parsed_3']

    sentence = text
    lemmatized_text = stemmer.stem(sentence)

    lemmatized_text_list.append(lemmatized_text)
df['title_parsed_4'] = lemmatized_text_list
```

Kode Sumber 4.13 Lemmatisasi Teks

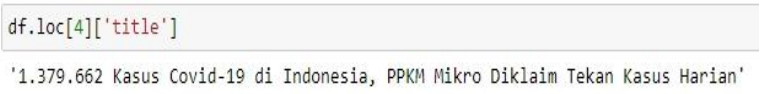
e. Menghapus *stopword*

```
STOPWORDS= stopwords.words('indonesian')
STOPWORDS.extend(['covid','covid-
19','covid19,','korona','corona','indonesia'])
stop_words = list(STOPWORDS)

df['title_parsed_5'] = df['title_parsed_4']

```

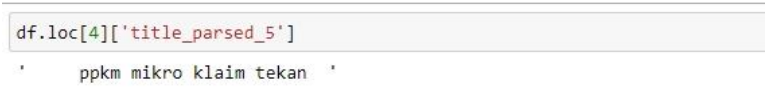
```
for stop_word in stop_words:
    regex_stopword = r"\b" + stop_word + r"\b"

    df['title_parsed_5'] =
df['title_parsed_5'].str.replace(regex_stopword,
'')
```

Kode Sumber 4.14 Menghapus *stopword*

Contoh hasil sebelum dan sesudah pemrosesan teks pada judul berita terdapat pada Gambar 4.1 dan Gambar 4.2.



Gambar 4.1 Sebelum Pemrosesan Teks



Gambar 4.2 Sesudah Pemrosesan Teks

4.3.3 Implementasi Rekayasa Fitur

Pada tahap ini hasil dari Klasifikasi Teks dilakukan klasifikasi teks menggunakan program dengan bahasa pemrograman Python. Hasil dari pemrosesan teks digunakan dalam pembuatan data label dan fitur. Implementasi tahap ini dapat dilihat pada Kode Sumber 4.16 dan Kode Sumber 4.17.

a. Pebagian Data Tes Latih dan Tes Set

```
df2 = df[list_columns]

df2 = df2.rename(columns={'title_parsed_6':
'title_parsed'})
```

Kode Sumber 4.15 Pembagian Latih – Set Tes

```
df2['label_code'] = df2['label']
df2 = df2.replace({'label_code':label_codes})

X_train, X_test, y_train, y_test =
train_test_split(df2['title_parsed'],
df2['label_code'], test_size=0.15, random_state=8)
```

Kode Sumber 4.16 Pembagian Latih – Set Tes

b. Representasi Teks

```
# Parameter election
ngram_range = (1,2)
min_df = 10
max_df = 1.
max_features = 300

tfidf = TfidfVectorizer(encoding='utf-8',
                        ngram_range=ngram_range,
                        stop_words=None,
                        lowercase=False,
                        max_df=max_df,
                        min_df=min_df,
                        max_features=max_features,
                        norm='l2',
                        sublinear_tf=True)
features_train =
tfidf.fit_transform(X_train).toarray()
features_train2 = tfidf.fit_transform(X_train)
sums = features_train2.sum(axis = 0)

labels_train = y_train

features_test = tfidf.transform(X_test).toarray()
labels_test = y_test
```

Kode Sumber 4.17 Representasi Teks

Hasil dari variable features_train, labels_train, features_test, labels_test dan tfidf disimpan untuk dilakukan pelatihan model.

4.3.4 Implementasi Pelatihan Model

Pada tahap ini penulis melakukan pelatihan model pada berbagai jenis model yaitu Random Forest, Support Vector Machine, K-Nearest Neighbor, dan Multinomial *Naïve* Bayes. Kelima model tersebut dilakukan uji perfoma dengan melakukan penyetelan hiperparameter sampai mendapatkan akurasi set latih. Setelah mendapatkan akurasi set latih , penulis membandingkan akurasi set latih tertinggi pada berbagai jenis model untuk dijadikan model utama. Hasil akurasi set latih pada setiap model dapat dilihat pada Gambar 4.3.

	Model	Training Set Accuracy
2	Random Forest	0.963004
3	SVM	0.930493
1	Multinomial Naïve Bayes	0.891256
0	KNN	0.857997

Gambar 4.3 Perbandingan Akurasi Set Latih Setiap Model

Dari Gambar 4.3 dapat dilihat bahwa model Random Forest memiliki akurasi set latih paling tinggi. Sehingga model Random Forest digunakan untuk mendapat atribut label pada data berita. Kode sumber dan penjelasan kode untuk melakukan proses pelatihan model Random Forest sebagai berikut:

1. Memuat data features_train, labels_train, pada proses rekayasa fitur. Implementasi tahap ini dapat dilihat pada Kode Sumber 4.18.

```
path_df = "C:/Users/asus-pc/Documents/PBA/Tugas
Akhir/Untitled Folder/Pickles_title/df.pickle"
with open(path_df, 'rb') as data:
    df = pickle.load(data)

path_features_train = "C:/Users/asus-
pc/Documents/PBA/Tugas Akhir/Untitled
Folder/Pickles_title/features_train.pickle"
with open(path_features_train, 'rb') as data:
    features_train = pickle.load(data)

path_labels_train = "C:/Users/asus-
pc/Documents/PBA/Tugas Akhir/Untitled
Folder/Pickles_title/labels_train.pickle"
with open(path_labels_train, 'rb') as data:
    labels_train = pickle.load(data)

```

Kode Sumber 4.18 Menambahkan Fitur Label

2. Melakukan pencarian hiperparameter terbaik dengan menggunakan *randomized searchcross validation* dan *grid search cross validation* dengan 3-*Fold Cross Validation* dengan 50 iterasi. Sebelum melakukan proses ini, penulis melakukan inisialisasi list variabel berupa parameter dari jumlah pohon dalam hutan, maksimum fitur dalam 1 node, jumlah level dalam satu pohon keputusan, jumlah minimum titik data pada node sebelum node dipisah, jumlah minimum titik data yang diizinkan dalam simpul daun dan metode pengambilan sampel titik data. List variabel tersebut pertama digunakan pada proses *randomized cross validation* untuk mendapatkan nilai hiperparameter pada setiap list variable. Implementasi tahap ini dapat dilihat pada Kode Sumber 4.19

```
n_estimators = [int(x) for x in np.linspace(start =
200, stop = 1000, num = 5)]
max_features = ['auto', 'sqrt']
max_depth = [int(x) for x in np.linspace(20, 100,
num = 5)]
max_depth.append(None)
min_samples_split = [2, 5, 10]
```

```
6.  min_samples_leaf = [1, 2, 4]
7.  bootstrap = [True, False]
8.  random_grid = {'n_estimators': n_estimators,
9.                 'max_features': max_features,
10.                'max_depth': max_depth,
11.                'min_samples_split':
    min_samples_split,
12.                'min_samples_leaf':
    min_samples_leaf,
13.                'bootstrap': bootstrap}


14. # First create the base model to tune
15. rfc = RandomForestClassifier(random_state=8)
16.
17. # Definition of the random search
18. random_search = RandomizedSearchCV(estimator = rfc,
19.                   param_distributions= random_grid,
20.                   n_iter=50,
21.                   scoring='accuracy',
22.                   cv=3,
23.                   verbose=1,
24.                   random_state=8)
25.
26. # Fit the random search model
27. random_search.fit(features_train, labels_train)
28. print(random_search.best_params_)
29. print(random_search.best_score_)
```

Kode Sumber 4.19 *Randomized Search Cross Validation*

```
{'n_estimators': 800, 'min_samples_split': 2, 'min_samples_leaf': 2, 'max_features': 'auto', 'max_depth': None, 'bootstrap': Fa
lse}
0.9215246636771299
```

Gambar 4.4 Hasil *Randomized Search Cross Validation*

Pada gambar 3.4 dapat dilihat nilai hyperparameter terbaik pada setiap list variable dan rata-rata akurasi model tersebut menggunakan *randomized search*. Nilai hiperparameter tersebut digunakan lagi dengan menggunakan metode *grid search cross validation* dan dibandingkan nilai akurasinya dengan *randomized search*. Implementasi program *Grid Search Cross Validation* dapat dilihat pada kode sumber dan output dari implementasi ini dapat dilihat pada Gambar 3.5. Implementasi tahap ini dapat dilihat pada Kode Sumber 4.20.

```
# Membuat parameter grid berdasarkan hasil
randomized search
bootstrap = [True]
max_depth = [70, 80, 90]
max_features = ['auto']
min_samples_leaf = [1, 2, 4]
min_samples_split = [5, 10, 15]
n_estimators = [400]
param_grid = {
    'bootstrap': bootstrap,
    'max_depth': max_depth,
    'max_features': max_features,
    'min_samples_leaf': min_samples_leaf,
    'min_samples_split': min_samples_split,
    'n_estimators': n_estimators
}
rfc = RandomForestClassifier(random_state=8)
cv_sets = ShuffleSplit(n_splits = 3, test_size =
.33, random_state = 8)

grid_search = GridSearchCV(estimator=rfc,
                           param_grid=param_grid,
                           scoring='accuracy',
                           cv=cv_sets,
                           verbose=1)

grid_search.fit(features_train, labels_train)
print(grid_search.best_params_)
print(grid_search.best_score_)
```

Kode Sumber 4.20 *Grid Search Cross Validation*

```
{'bootstrap': True, 'max_depth': 70, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 40
0}
0.92420814479638
```

Gambar 4.5 Hasil *Grid Search Cross Validation*

Dengan membandingkan akurasi model *Randomized Search Cross Validation* dan *Grid Search Cross Validation* pada Gambar 4.4 dan Gambar 4.5, dapat disimpulkan bahwa metode *Grid Search Cross Validation* digunakan sebagai metode penyetelan hiperparameter dalam menguji performa menggunakan model *Random Forest*. Implementasi pengujian performa akurasi data

latih menggunakan model *Random Forest* dengan metode *Grid Search Cross Validation* dapat dilihat pada Kode Sumber 4.21. Hasil dari pengujian performa berupa akurasi set latih pada setiap Model dapat dilihat pada Gambar 4.3 bersamaan dengan akurasi set latih menggunakan model lain. Sedangkan Kode Sumber model lain dapat dilihat melalui lampiran.

```
1. best_rfc =  grid_search.best_estimator_
2. best_rfc.fit(features_train, labels_train)
3. print(accuracy_score(labels_train,
   best_rfc.predict(features_train)))
```

Kode Sumber 4.21 Peforma akurasi Data latih

4.3.4 Implementasi Menambah Atribut Label

Selain menggunakan model *Random Forest* dalam proses mendapatkan atribut label pada dataset berita, juga menggunakan file tfidf dari representasi teks pada subbab . Impementasi program dan penjelasan program untuk menambah atribut label dijabarkan sebagai berikut.

4.3.4.1 Implementasi Fungsi Utama

Program diawali dengan memuat dataset berita yang tidak memiliki atribut label dan file model klasifikasi *Random Forest* dan tfidf. Setiap data berita, dilakuakan prediksi label yang paling cocok pada data tersebut menggunakan judul berita pada setiap data. Nama label yang digunakan sesuai pada Tabel 3.4. Akan tetapi, apabila tidak ada label yang cocok atau kecocokan label dengan judul berita kurang dari 65%. Maka atribut label akan dilabelkan “lain-lain”. Hasil prediksi label disimpan kedalam database MySQL. Implementasi program fungsi utama dapat dilihat pada Kode Sumber 4.22.

```
1.  mycursor.execute("SELECT title FROM news where
    label is null or label=''")
2.  data_berita = mycursor.fetchall()
3.
4.  path_rfc = path_models + 'best_rfc.pickle'
5.  with open(path_rfc, 'rb') as  data:
6.      rfc_model = pickle.load(data)
7.
8.  with open(path_tfidf, 'rb') as data:
9.      tfidf = pickle.load(data)
10.
11. kode_label = {
12.     'informasi': 0,
13.     'donasi': 1,
14.     'kritik': 2,
15.     'hoaks': 3,
16. }
17.
18. for baris in data_berita:
19.   nama_label,label_persen =
    prediksi_dari_teks(baris[0])
20.   if(label_persen>=65):
21.     sql = "UPDATE news SET label = '"+nama_label+
    "' WHERE title = '"+str(baris[0])+"'"
22.     mycursor.execute(sql)
23.     mydb.commit()
24.   else:
25.     sql = "UPDATE news SET label = '"+"lain-lain"+
    "' WHERE title = '"+str(baris[0])+"'"
26.     mycursor.execute(sql)
27.     mydb.commit()
```

Kode Sumber 4.22 Menambahkan Atribut Label

4.3.4.2 Implementasi Fungsi Membuat Fitur Teks

Subbab ini menjelaskan fungsi untuk mendapatkan fitur dari setiap judul berita dengan menggunakan pemrosesan teks. Implementasi program fungsi membuat fitur dapat dilihat pada Kode Sumber Kode Sumber 4.23.

```
1.  d def membuat_fitur(text):
2.
3.      lemmatized_text_list = []
4.      df = pd.DataFrame(columns=['title'])
5.      df.loc[0] = text
6.      df['title_parsed_1'] =
    df['title'].str.replace("\r", " ")
7.      df['title_parsed_1'] =
    df['title_parsed_1'].str.replace("\n", " ")
8.      df['title_parsed_1'] =
    df['title_parsed_1'].str.replace("    ", " ")
9.      df['title_parsed_1'] =
    df['title_parsed_1'].str.replace('"', '')
10.     df['title_parsed_2'] =
    df['title_parsed_1'].str.lower()
11.     df['title_parsed_3'] = df['title_parsed_2']
12.
13.     punctuation_signs = list("?:!.,;")
14.     for punct_sign in punctuation_signs:
15.         df['title_parsed_3'] =
    df['title_parsed_3'].str.replace(punct_sign, '')
16.     factory = StemmerFactory()
17.     stemmer = factory.create_stemmer()
18.     text = df.loc[0]['title_parsed_3']
19.     sentence = text
20.     lemmatized_text = stemmer.stem(sentence)
21.     lemmatized_text_list.append(lemmatized_text)
22.     df['title_parsed_4'] = lemmatized_text_list
23.     df['title_parsed_5'] = df['title_parsed_4']
24.
25.     punctuation_signs = list("?:!.,;")
26.     STOPWORDS= stopwords.words('Indonesian')
27.     STOPWORDS.extend(['covid','covid-19','covid-
    19,','korona','2020','corona',
    'corona,','2021','0','1','2','3','4','5','6','7',
    '8','9','10','11','12','13','14','15','16','17','
    18','19','20','21','22','23','24','25','26','27',
    '28','29','30','31','ribu','juta','-'])
28.     stop_words = list(STOPWORDS)
29.     for stop_word in stop_words:
30.         regex_stopword = r"\b" + stop_word +
    r"\b"
31.         df['title_parsed_5'] =
    df['title_parsed_5'].str.replace(regex_stopword,
    '')
32.     df['title_parsed_5'] =
    df['title_parsed_5'].str.replace('-', '')
33.     df = df.rename(columns={'title_parsed_5':
    'title parsed'})
```

```
34.     df = df['title_parsed']
35.     #df = df.rename(columns={'title_parsed_6':
    'title_parsed'})
36.
37.     # TF-IDF
38.     features = tfidf.transform(df).toarray()
39.
40.     return features
```

Kode Sumber 4.23 Fungsi Membuat Fitur Teks

4.3.4.3 Implementasi Fungsi Mendapatkan Nama Label

Subbab ini menjelaskan fungsi untuk mendapatkan label pada setiap judul berita berdasarkan kodel label yang diberikan sebelumnya. Implementasi program fungsi mendapatkan nama label dapat dilihat pada Kode Sumber 4.24.

```
1.  def get_nama_label(label_id):
2.      for label, id_ in kode_label.items():
3.          if id_ == label_id:
4.              return label
```

Kode Sumber 4.24 Fungsi Mendapatkan Nama Label

4.3.4.4 Implementasi Fungsi Memprediksi Label Dari Teks

Subbab ini menjelaskan fungsi untuk melakukan prediksi label dari setiap judul berita dengan menggunakan model *Random Forest*. Tahap pertama adalah mendapatkan fitur dari judul berita dengan memmanggil fungsi fitu teks. Hasil dari fitur teks akan digunakan untuk memprediksi label yang cocok Implementasi program fungsi mendapatkan nama label dapat dilihat pada Kode Sumber Kode Sumber 4.25.

```
1.  def prediksi_dari_teks(teks):
2.      prediksi_rfc =
    rfc_model.predict(fitur_teks(teks))[0]
3.      prediksi_rfc_proba =
    rfc_model.predict_proba(fitur_teks(teks))[0]
4.
5.      label_rfc = get_nama_label(prediksi_rfc)
6.      return label_rfc,prediksi_rfc_proba.max()*100
```

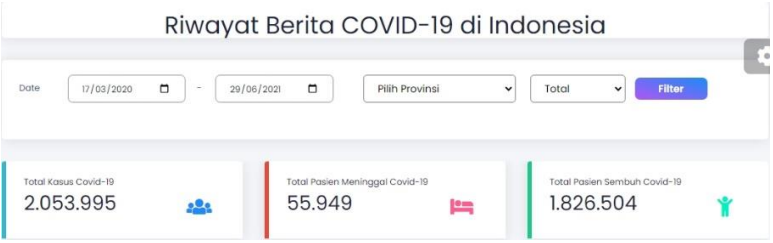
Kode Sumber 4.25 Fungsi Prediksi Label dari Teks

4.4 Implementasi Visualisasi Web

Subbab ini menjelaskan hasil dari dataset COVID-19 dan berita setelah klasifikasi teks yang diimplementasikan kedalam web. Framework yang digunakan dalam proses pembuatan web adalah Laravel. Pada subbab ini dijelaskan dan ditampilkan halaman HTML dengan rancangan halaman antarmuka yang terdapat pada Bab III.

4.4.1 Halaman Pencarian Berita COVID-19

Halaman utama riwayat berita COVID-19 merupakan halaman yang pertama kali muncul sebelum menuju halaman statistik berita atau daftar berita. Halaman ini menampilkan grafik COVID-19 dan berita di seluruh Indonesia dari tanggal 18 Maret 2020 sampai tanggal saat membuka web. Pengguna bisa menyaring data COVID-19 dan berita berdasarkan tanggal, tempat dan data harian atau total. Penyaringan data bisa dilakukan dengan mengubah filter judul COVID-19. Tampilan Halaman Pencarian Berita dapat dilihat pada Gambar 4.6,Gambar 4.7 dan Gambar 4.8.



Gambar 4.6 Tampilan Formulir dan Data COVID-19

Gambar 4.7 Tampilan Grafik Total Kasus COVID-19 dan Berita di Indonesia

Gambar 4.8 Tampilan Grafik Peningkatan Kasus COVID-19 dan Berita di Provinsi Jawa Timur

4.4.2 Halaman Daftar Riwayat Berita COVID-19

Halaman daftar riwayat berita COVID-19 merupakan halaman yang menampilkan setiap berita di seluruh provinsi dan kota. Tampilan awal halaman daftar riwayat berita adalah formulir filter, 4 data berita terbaru, diikuti dengan daftar berita lainnya pada bagian selanjutnya yang dibagi menjadi 10 data berita untuk

setiap halaman. Apabila memilih berita provinsi, maka memunculkan halaman list kota dalam provinsi tersebut sesuai Gambar 4.9. Setiap kota memiliki tampilan yang berbeda, tegantung dari jumlah berita pada kota tersebut. Tampilan Halaman Daftar Berita dapat dilihat pada Gambar 4.10 dan Gambar 4.11.

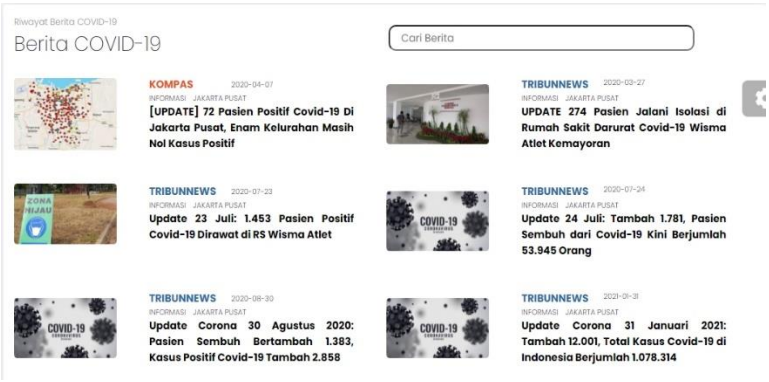
Gambar 4.9 Tampilan Formulir dan List Daerah Jakarta

Gambar 4.10 Tampilan Daftar Berita Terbaru Daerah Jakarta Pusat

Gambar 4.11 Tampilan Daftar Berita Kota Jakarta Pusat

4.4.3 Halaman Pencarian Berita

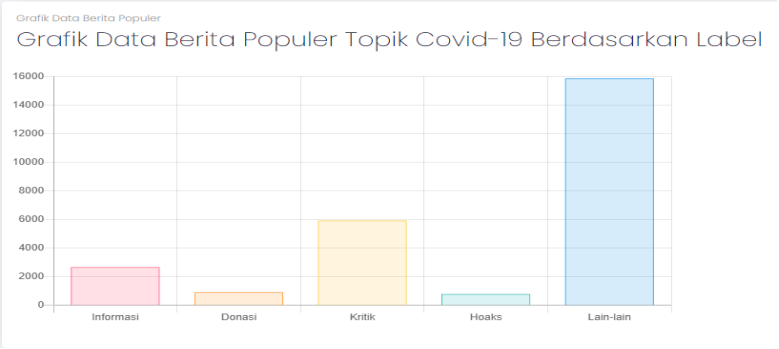
Halaman ini digunakan untuk menampilkan daftar riwayat berita COVID-19 berdasarkan hasil pencarian judul berita pada halaman daftar riwayat berita. Tampilan Halaman Pencarian Berita dapat dilihat pada Gambar 4.12.



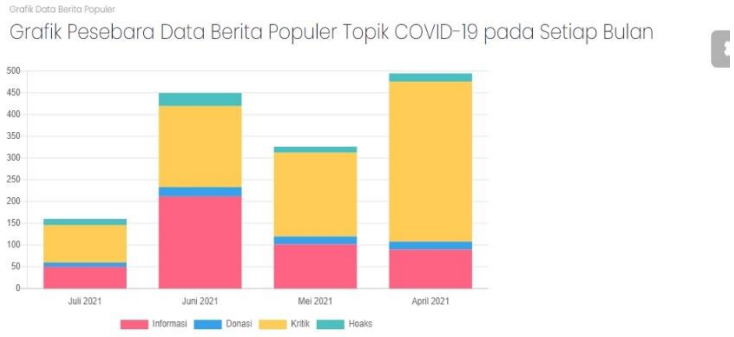
Gambar 4.12 Tampilan Pencarian Judul Berita dengan kata kunci "Update" di Kota Jakarta Pusat

4.4.4 Halaman Statistik Berita

Halaman ini digunakan untuk menampilkan statistik riwayat berita COVID-19 berdasarkan label. Statistik berupa *bar chart* dan *stacked bar chart. Bar chart* menampilkan jumlah berita berdasarkan kategori berita. Tampilan *bar chart* dapat dilihat padaGambar 4.13. Sementara *stacked bar chart* menampilkan pesebaran kategori berita pada setiap bulannya. Tampilan *stacked bar chart* dapat dilihat pada Gambar 4.14.



Gambar 4.13 Jumlah Berita berdasarkan Label



Gambar 4.14 Jumlah Pesebaran Label Berita pada bulan April 2021- Juli 2021

[*Halaman ini sengaja dikosongkan*]

BAB V
UJI COBA DAN EVALUASI

Bab ini membahas mengenai hasil, uji coba dan evaluasi dari implementasi yang telah dilakukan pada tugas akhir ini. Hasil yang didapatkan dari proses uji coba akan dievaluasi sehingga dapat diambil kesimpulan untuk kebutuhan bab selanjutnya.

5.1 Lingkungan Uji Coba

Pada tugas akhir ini, digunakan beberapa perangkat serta pustaka yang berfungsi untuk proses uji coba. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dapat dilihat pada Tabel 5.1.

Tabel 5.1 Spesifikasi Lingkungan Implementasi

No.	Jenis Perangkat	Spesifikasi
1	Perangkat Keras	• *Processor*: Intel(R) Core(TM) i7-7200U CPU @ 2.50GHz • *Random Access Memory*: 8GB
2	Perangkat Lunak	• *Operating System*: Windows 10 Home Single Language • Bahasa pemrograman Python 3.8.5 64-bit • *Integrated Development Environment*: Jupyter Notebook • *Text Editor* Sublime Text 3 • *Google Form*

5.2 Pengujian Klasifikasi Teks

Pada subbab ini dijelaskan hasil dari tiap tahapan proses teks klasifikasi menggunakan dataset sampel berita dan skenario

pengujian yang dilakukan pada setiap tahapnya. Tahapan pengujian dibagi menjadi dua yaitu tahapan pengujian kesesuaian kata kunci pada setiap label, dan pengujian peforma setiap model.

5.2.1 Pengujian Fitur Label Berita

Pengujian fitur dilakukan agar setiap label mempunyai kata kunci yang mempunyai hubungan dengan nama label masing-masing. Selain itu, Pengujian ini dilakukan agar bisa menguji fitur data latih agar membantu dalam proses pelatihan model. Kesesuaian kata kunci menggunakan dua cara yaitu mencari kata unigram dan bigram yang paling cocok.

Unigram dan bigram adalah bagian dari n-gram , yaitu sebuah metode pengolahan dokumen dengan membagi kata sebanyak n sesuai urutan teks. Dalam pengujian kali ini selain membantu meningkatkan efektifitas klasifikasi, penulis menggunakan n_gram untuk mencari kesesuaian kalimat pada setiap label. Kalimat tersebut memiliki maksimal n buah kata. Pada kasus ini, penulis mencari kalimat yang memiliki satu (unigram) dan 2 (bigram). Proses mencari unigram menggunakan data yang telah dipasang dan dirubah menjadi sebuah fitur menggunakan metode TFIDF. Agar hasil unigram dan bigram sesuai dengan label, penulis hanya mengambil 5 unigram dan bigram yang paling sesuai. Hasil unigram dan bigram pada setiap label dapat dilihat pada Tabel 5.2, Tabel 5.3, Tabel 5.4, dan Tabel 5.5.

Tabel 5.2 Unigram dan Bigram Label Informasi

No	Unigrams	Bigrams
1	Update	Positif sembuh
2	Total	Total positif
3	Sembuh	Sembuh tinggal
4	Positif	Update februari
5	Bantu	Update januari

Tabel 5.3 Unigram dan Bigram Label Donasi

No	Unigrams	Bigrams
1	Bantu	Salur bantu
2	Sumbang	Bantu tangan
3	Donasi	Bantu warga
4	Ikan	Terima bantu
5	Dampak	Warga dampak

Tabel 5.4 Unigrams dan Bigrams Label Kritik

No	Unigrams	Bigrams
1	Gagal	Sembuh tinggal
2	Bukti	Total Positif
3	Kritik	Positif sembuh
4	Singgung	Pemprov dki
5	Buruk	Pasien Positif

Tabel 5.5 Unigrams dan Bigrams Hoaks

No	Unigrams	Bigrams
1	Fakta	Positif sembuh
2	Klaim	Total Positif
3	Hoaks	Sembuh Tinggal
4	Klarifikasi	Wali Kota
5	Update	Update Februari

Dari hasil tabel diatas dapat dilihat bahwa setiap label memiliki korelasi bigrams yang sama kecuali label donasi, sedangkan untuk unigrams sama sekali hampir tidak ada label yang memiliki unigram yang sama.

Jika dilakukan analisa pada unigrams di masing-masing label, label donasi memiliki kata “ikan” yang tidak ada hubungannya dengan donasi. Tetapi setelah diteliti lagi, kata ikan berasal dari kata “berikan” yang telah dilakukan proses

lemmatilisasi teks. Sedangkan pada label kritik dan hoaks, seharusnya salah satunya memiliki kata "pemerintah" yang merupakan adalah aktor utama dalam pengendalian covid-19. Tetapi setelah diteliti lagi, kata pemerintah telah di lemmatilisasi teks menjadi kata "perintah" dan kata "perintah" termasuk kata yang dihapus pada proses penghapusan *stopword* karena termasuk dalam list *stopword* NLTK Bahasa Indonesia.

5.2.2 Pengujian Peforma Model

Pengujian peforma dilakukan untuk menganalisa kesesuaian label dengan judul berita. Pengujian peforma dilakukan dengan menggunakan data tes diambil dari 15% dari sampel data berita.

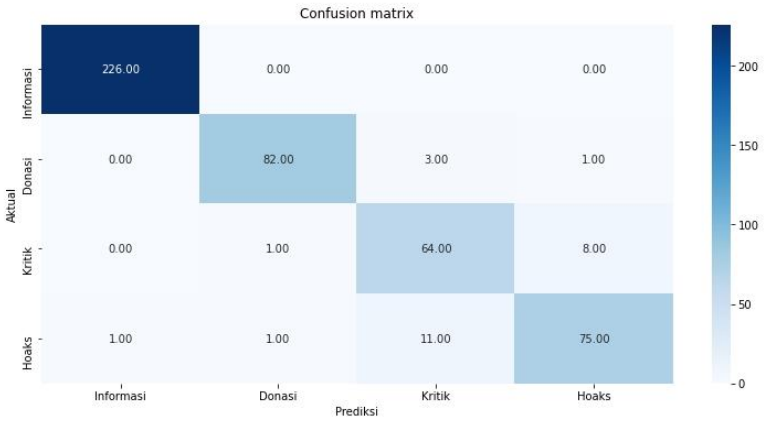
Pada pengujian ini, setiap model mengukur peforma data tes dengan mencari nilai *precision,recall, f-1 score* menggunakan 3-*fold cross validation.* Selain itu, agar dapat mengetahui kinerja dari setiap model klasifikasi dan pesebaran hasil prediksi label untuk setiap model, penulis menggunakan *confusion matrix* dalam mendukung hasil pengujian peforma. Hasil dari *confusion matrix* dianalisa untuk didapatkan kesesuai label dengan judul berita.

5.2.2.1 Random Forest

Pengujian Peforma klasifikasi menggunakan *Random Forest* dapat dilihat pada Tabel 5.6. Sedangkan *confusion matrix Random Forest* dapat dilihat pada Gambar 5.1.

Tabel 5.6 Pengujian Peforma *Random Forest*

Label	**Precission**	**Recall**	**F1-Score**
Informasi	1.0	1.0	1.0
Donasi	0.98	0.94	0.96
Kritik	0.80	0.89	0.84
Hoaks	0.89	0.83	0.86
Rata-Rata	0.92	0.92	0.91



Gambar 5.1 *Confusion Matrix Random Forest*

Dari hasil tabel dapat dilihat bahwa data label Informasi memiliki peforma maksimal dari segi *precission*, *recall* dan *F1-score*. Peforma data label donasi memiliki peforma yang tinggi. Tetapi data label kritik dan hoaks memiliki peforma yang rendah.

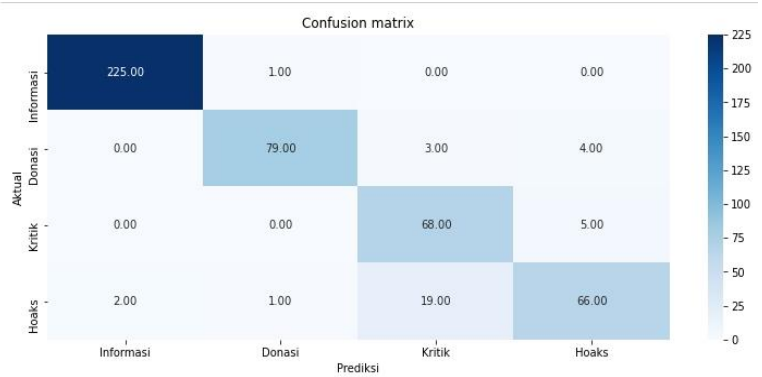
Apabila dilihat dari *confusion matrix*, kebanyakan pesebaran data label kritik yang prediksinya salah berada pada label hoaks dan begitu sebaliknya. Ini dapat disebabkan karena hasil pemrosesan teks judul berita pada data actual hoaks memiliki kesesuain yang tinggi pada label kritik sehingga *decision tree* memutuskan memilih label kritik. Selain itu dari data label donasi,kritik dan hoaks tidak ada yang mendapatkan hasil prediksi pada label informasi meskipun memiliki label informasi

5.2.2.2 Support Vector Machine (SVM)

Pengujian Peforma klasifikasi menggunakan SVM dapat dilihat pada Tabel 5.7 Pengujian Peforma SVM. Sedangkan *confusion matrix* SVM dapat dilihat pada Gambar 5.2.

Tabel 5.7 Pengujian Peforma SVM

Label	**Precission**	**Recall**	**F1-Score**
Informasi	0.99	1.0	0.99
Donasi	0.98	0.92	0.95
Kritik	0.76	0.93	0.83
Hoaks	0.88	0.75	0.81
Rata-Rata	0.92	0.92	0.91



Gambar 5.2 *Confusion Matrix SVM*

Dari hasil tabel dapat dilihat bahwa data label Informasi memiliki peforma maksimal dari segi *precission*, *recall* dan *F1-score*. Peforma data label donasi memiliki peforma yang tinggi. Tetapi data label kritik dan hoaks memiliki peforma yang rendah. Meskipun memiliki peforma yang rendah, data label kritik memiliki nilai *recall* yang tinggi. Nilai ini menandakan bahwa tingginya rasio data yang memiliki label kritik dibandingkan dengan keseluruhan data yang diprediksikan memiliki label kritik.

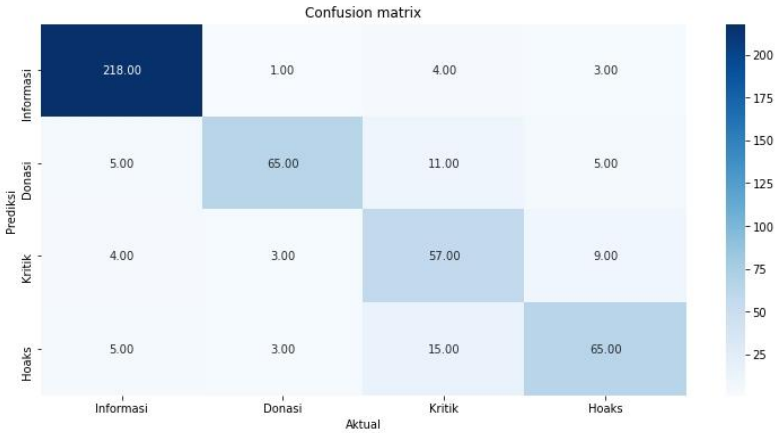
Apabila dilihat dari *confusion matrix*, kebanyakan pesebaran data label hoaks yang prediksinya salah berada pada label kritik dan begitu juga sebaliknya. Ini bisa dsebabkan karena label hoaks memiliki kombinasi kata bigram dan unigram dengan label kritik.

5.2.2.3 K-Nearest Neighbors

Pengujian Peforma klasifikasi menggunakan KNN dapat dilihat pada Tabel 5.8. Sedangkan *confusion matrix* KNN dapat dilihat pada Gambar 5.3.

Tabel 5.8 Pengujian Peforma KNN

Label	Precission	Recall	F1-Score
Informasi	0.91	0.98	0.94
Donasi	0.86	0.74	0.80
Kritik	0.73	0.77	0.75
Hoaks	0.77	0.69	0.73
Rata-Rata	0.82	0.80	0.80



Gambar 5.3 *Confusion Matrix KNN*

Dari hasil Tabel 5.8 dapat dilihat bahwa data label Informasi memiliki peforma maksimal dari segi *precission*, *recall* dan *F1-score*. Peforma data label donasi memiliki peforma yang lumayan bagus. Tetapi data label kritik dan hoaks memiliki peforma yang rendah.

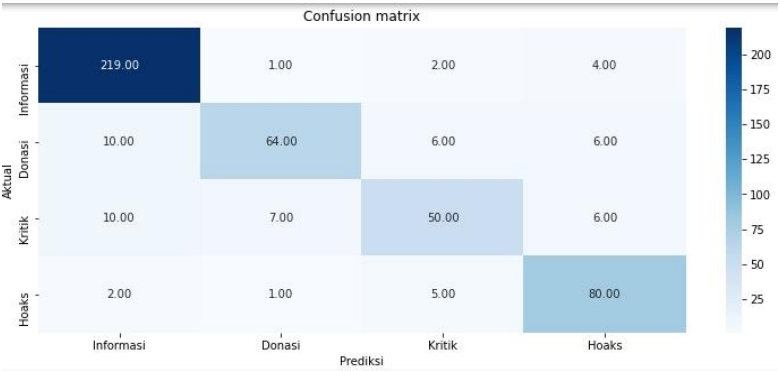
Apabila dilihat dari Gambar 5.3, kebanyakan pesebaran data label informasi,donasi, dan hoaks yang prediksinya salah berada pada label kritik. Label kedua yang memiliki pesebaran data label yang prediksinya salah adalah label informasi. Label informasi meskipun memiliki data yang banyak tetapi pesebaran data yang memiliki prediksi salah sangat sedikit dibandingkan label lain. Peforma rendah pada model klasifikasi KNN dapat disebabkan karena banyaknya sampel data sehingga menyebabkan prediksi data label yang salah.

5.2.2.4 Multinomial *Naïve* Bayes

Pengujian Peforma klasifikasi menggunakan KNN dapat dilihat pada Tabel 5.9. Sedangkan *confusion matrix* KNN dapat dilihat pada Gambar 5.4.

Tabel 5.9 Pengujian Peforma *Naïve Bayes*

Label	Precission	Recall	F1-Score
Informasi	0.91	0.97	0.94
Donasi	0.88	0.74	0.81
Kritik	0.79	0.68	0.74
Hoaks	0.83	0.91	0.87
Rata-Rata	0.85	0.83	0.84



Gambar 5.4 *Confusion Matrix Naïve Bayes*

Dari hasil Tabel 5.9 dapat dilihat bahwa data label Informasi memiliki peforma maksimal dari segi *precission*, *recall* dan *F1-score*. Peforma data label donasi memiliki peforma yang lumayan bagus. Tetapi data label kritik dan hoaks memiliki peforma yang rendah..

Apabila dilihat dari Gambar 5.4, kebanyakan pesebaran data label donasi,kritik dan hoaks yang prediksinya salah berada pada label informasi. Ini dapat disebabkan karena probabilistas kemunculan unigram dan bigram lebih banyak muncul pada label informasi sehingga dapat disebabkan.

5.2.3 Pengujian Penentuan Model

Pengujian yang terakhir adalah pengujian untuk menentukan model klasifikasi data berita pada tugas akhir dengan menggunakan data tes. Pengujian dilakukan dengan membandingkan akurasi set tes pada setiap model untuk menentukan model terbaik dalam proses teks klasifikasi. Pebandingan ini dilakukan untuk membuktikan apakah model *Random Forest* tetap menjadi model terbaik dalam proses teks klasifikasi dengan menggunakan data latih. Hasil dari pembandingan akurasi set tes dapat dilihat pada Gambar 5.5.

	Model	Test Set Accuracy
2	Random Forest	0.945032
3	SVM	0.926004
1	Multinomial Naïve Bayes	0.873150
0	KNN	0.849894

Gambar 5.5 Perbandingan Akurai Set Tes Setiap Model

Dari Gambar diatas, dapat dilihat tidak ada perubahan ranki ng model terbaik berdasarkan akurasi set tes. Sehingga disimpulkan bahwa model Random Forest merupakan model yang cocok da lam klasifikasi data berita pada tugas akhir ini.

5.3 Pengujian Website

Pengujian ini dilakukan untuk menguji hasil visualisasi data COVID-19 dan berita yang ditampilkan dalam web sistem informasi COVID-19. Pengujian dilakukan oleh pengguna web dengan membuat form survei kepada 50 orang. Isi survei terdiri dari profil responden yang terdiri dari nama, provinsi dan kota tempat tinggal pengguna dan media pengguna dalam, kebiasaan pengguna dalam membaca berita mengenai COVID-19. Kriteria responden dalam mengisi survei dalam mengisi survei dapat dilihat pada Tabel 5.10.

Tabel 5.10 Kriteria Responden yang Mengisi Survei

No	Kriteria Responden
1	51 Responden terdiri dari pelajar, mahasiswa dan pekerja dengan usia rata-rata 22,5 tahun.
2	32% responden berasal dari Jawa Timur , 44% dari Provinsi di Jawa (non-Jawa Timur), dan 24% dari provinsi luar Jawa.
3	32% responden sering membaca berita terkait COVID-19 50% kadang-kadang dan 18% jarang/ tidak pernah membaca berita.
4	90% responden membaca berita terkait COVID-19 melalui media sosial, 72% melalui portal berita online dan 16% melalui grup chat seperti WA/Line, 16% melalui koran.
5	88% responden membaca berita terkait COVID-19 di daerah tempat tinggal melalui media sosial, 60% melalui

	portal berita online dan 25% melalui grup chat seperti WA/Line, 30% melalui Koran.

Selama pengujian web oleh responden, responden dipersilahkan menjelajahi web sistem informasi COVID-19. Selain itu penulis juga melakukan survei pemahaman responden terkait visualisasi data berita, serta pendapat dan saran responden mengenai web. Hasil dari form tersebut dijadikan bahan evaluasi oleh penulis saat proses tugas akhir berlansung. Pertanyaan dan jawaban responden terkait pengujian web dapat dilihat pada tabel

Tabel 5.11 Kriteria Responden yang Mengisi Survei

No	Pertanyaan	Skor (%)			
		1	2	3	4
1	Apakah berita COVID-19 di provinsi/kota Anda sesuai dengan apa yang ditampilkan di web?	0	0	66	34
2	Apakah menurut anda halaman statistik berita diperlukan?	0	0	2	98
3	Apakah memahami maksud dari halaman statistik berita?	0	16	58	26
4	Bagaimana nilai Anda terkait web sistem informasi berita COVID-19 di Indonesia?	0	8	44	48

Selain itu responden juga memberikan saran dan kritik terkait web sistem informasi COVID-19. Dari semua jawaban responden, penulis melakukan evaluasi terhadap sistem agar sistem dapat bekerja lebih baik kedepannya dan bermanfaat bagi pengguna terutama dalam penyampaian berita COVID-19 di setiap daerah.

[*Halaman ini sengaja dikosongkan*]

BAB VI
KESIMPULAN

Bab ini membahas mengenai kesimpulan yang diperoleh dari pengerjaan tugas akhir dan saran terkait pengembangan dari tugas akhir ini yang dapat dilakukan di masa yang akan mendatang.

6.1. Kesimpulan

Berdasarkan penjabaran di bab-bab sebelumnya, dapat disimpulkan beberapa poin terkait proses visualisasi dan implementasi kedalam web sistem informasi

1. Proses *web scraping* untuk mengambil halaman portal berita dan mengambil data kawalcovid-19 berhasil dilakukan. Proses pengambilan data berita dilakukan dengan mengambil data dari halaman tersebut dengan mengambil elemen HTML pada halaman web yang dituju. Sedangkan proses pengambilan data covid-19 adalah mencari data tabel berdasarkan lokasi *sheet* dan *cell* pada halaman *spreadsheet*.
2. Model yang digunakan pada klasifikasi teks pada data berita adalah Random Forest. Random Forest dipilih karena memiliki akurasi data latih dan akurasi data tes yang paling tinggi dibandingkan model lain. Random Forest juga dipilih karena data sampel yang dimiliki sangat besar dan pesebaran jumlah data pada setiap label tidak merata.
3. Meskipun memiliki akurasi data latih dan data tes diatas 92%. Pesebaran data label berita lebih banyak mengarah ke label kritik. Label kritik memiliki kombinasi kata bigram yang sama dan banyak dengan label informasi dan hoaks. Selain itu pesebaran data label kritik menggunakan model *Random Forest* sangat tinggi dibandingkan dengan model lain.

4. Berdasarkan hasil survei oleh 50 responden, tampilan berita covid-19 sudah sesuai dengan apa yang ditampilkan kedalam website. Selain itu, pengguna merasa terbantu dengan adanya web berita topik COVID-19.

6.2. Saran

Pada tugas akhir kali ini tentunya terdapat kekurangan serta nilai-nilai yang dapat penulis ambil. Berikut adalah saran-saran yang dapat digunakan untuk pengembangan di masa yang akan datang. Saran-saran ini didasarkan pada hasil desain implementasi dan uji coba yang telah dilakukan.

1. Memperbaiki sampel data berita untuk meningkatkan kualitas model saat proses klasifikasi teks.
2. Menambah jenis label berita untuk membedakan berbagai jenis berita terkait COVID-19.
3. Data berita COVID-19 bukan berasal dari portal berita online saja, tetapi juga bisa berasal dari informasi dari pemerintah pusat/daerah.
4. Menambahkan visualisasi atau analisa terkait COVID-19 pada Web Sistem Informasi COVID-19 sehingga sistem informasi ini dapat membantu terkait penanggulangan COVID-19.

[*Halaman ini sengaja dikosongkan*]

DAFTAR PUSTAKA

[1] W. J. Wiersinga and H. C. Prescott, "What Is COVID-19?," *JAMA - J. Am. Med. Assoc.*, vol. 324, no. 8, p. 816, 2020, doi: 10.1001/jama.2020.12984.

[2] H. A. Rothan and S. N. Byrareddy, "The epidemiology and pathogenesis of coronavirus disease (COVID-19) outbreak," *J. Autoimmun.*, vol. 109, no. February, p. 102433, 2020, doi: 10.1016/j.jaut.2020.102433.

[3] Worldometer, "COVID-19 Coronavirus Pandemic." https://www.worldometers.info/coronavirus/ (accessed Jan. 24, 2021).

[4] A. Azikin, Karno, P. Nurhascaryani, Fitriani, Suhaeti, and Y. Cahyono, "Indonesian government dilematics in Covid-19 pandemic handling," *Eur. J. Mol. Clin. Med.*, vol. 7, no. 7, pp. 125–133, 2020.

[5] Satgas Penanganan COVID-19, "Peta Sebaran COVID-19 Indonesia." https://kawalcovid19.id/ (accessed Jan. 27, 2021).

[6] S. Anggarini, "Fenomena Dalam Berita Covid-19," *J. Audience*, vol. 3, no. 2, pp. 224–249, 2020, doi: 10.33633/ja.v3i2.3628.

[7] K. Jahanbin and V. Rahmanian, "Using twitter and web news mining to predict COVID-19 outbreak," *Asian Pac. J. Trop. Med.*, vol. 13, no. 8, pp. 378–380, 2020, doi: 10.4103/1995-7645.279651.

[8] Akudigital, "13 Portal Berita Online Terbaik di Indonesia." https://www.akudigital.com/bisnis-tips/13-portal-berita-online-terbaik-di-indonesia/ (accessed Jan. 27, 2021).

[9] B. Zhao, "Web Scraping," no. December, 2018, doi: 10.1007/978-3-319-32001-4.

[10] A. V Saurkar, "An Overview On Web Scraping Techniques And Tools," pp. 363–367, 2018

[11] S. A. Salloum, M. Al-Emran, A. A. Monem, and K. Shaalan, "Using text mining techniques for extracting information

from research articles," *Stud. Comput. Intell.*, vol. 740, no. January, pp. 373–397, 2018, doi: 10.1007/978-3-319-67056-0_18.

[12] C. Hu, Y. Li, Y. Wang, and L. Wu, "Analysis of Hot News Based on Big Data," *Proc. - 17th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2018*, pp. 678–681, 2018, doi: 10.1109/ICIS.2018.8466427.

[13] K. Doshi, S. Gokhale, H. Mamtora, and P. Bide, "Analytics and Visualization of Trends in News Articles," *2019 6th IEEE Int. Conf. Adv. Comput. Commun. Control. ICAC3 2019*, 2019, doi: 10.1109/ICAC347590.2019.9036812.

[14] A. Ridok and R. Latifah, "Klasifikasi Teks Bahasa Indonesia Pada Corpus Tak Seimbang Menggunakan NWKNN," *Konf. Nas. Sist. dan Inform. 2015*, no. Oktober, pp. 222–227, 2015.

[15] M. F. Zafra, "Text Classification in Python," 2019. https://towardsdatascience.com/text-classification-in-python-dd95d264c802 (accessed Mar. 12, 2020).

[16] A. Pretorius, S. Bierman, and S. J. Steel, "A meta-Analysis of research in random forests for classification," *2016 Pattern Recognit. Assoc. South Africa Robot. Mechatronics Int. Conf. PRASA-RobMech 2016*, no. November 2016, 2017, doi: 10.1109/RoboMech.2016.7813171.

[17] M. Schott, "Random Forest Algorithm for Machine Learning," 2019. https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb (accessed Jun. 06, 2021).

[18] A. S. Nugroho, A. B. Witarto, and D. Handoko, "Support Vector Machine - Teori dan Aplikasinya dalam Bioinformatika," 2003.

[19] M. Rivki and A. M. Bachtiar, "Implementasi Algorimat K-Nearest Neighbor Dalam Pengklasifian Follower Twitter yang Menggunakan Bahasa Indonesia" 2017, doi: 10.21609/jsi.v13i1.500.

[20] A. Saleh, “Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga,” *Creat. Inf. Technol. J.*, vol. 2, no. 3, pp. 207–217, 2015.
[21] B. Liu, *Web Data Mining Exploring Hyperlinks, Contents, and Usage Data, Second Edition, New York: Spinge*. 2011.
[22] Laravel, “Laravel Documentation.” https://laravel.com/docs/8.x (accessed Mar. 01, 2021).

[*Halaman ini sengaja dikosongkan*]

LAMPIRAN

1. Kode Sumber Kata Kunci untuk Menambah Atribut Provinsi

```
1.  jawa_timur = ["jawa timur","jatim", "khofifah"]
2.  jawa_tengah = ["jawa tengah","ganjar", "jateng"]
3.  dki_jakarta = ["dki jakarta", "jakarta", "anies"]
4.  di_yogyakarta = ["di yogyakarta",
    "jogja","hamengkubuwana"]
5.  jawa_barat= ["jawa barat","jabar","jabar","ridwan
    kamil"]
6.  banten = ["banten", "serang", "wahidin halim"]
7.  aceh= ["aceh","aceh","nova iriansyah"]
8.  sumatera_barat = ["sumatera
    barat","sumbar","mahyeldi ansharullah"]
9.  sumatera_utara = ["sumatera utara","sumut", " edy
    rahmayadi"]
10. sumatera_selatan = ["sumatera selatan","sumsel",
    "herman deru"]
11. riau= ["riau", "riau", "syamsuar"]
12. kep_riau= ["kepulauan riau", "kepri", "Ansar
    Ahmad"]
13. lampung= ["lampung", "lampung", "arinal
    djunaidi"]#12
14. jambi= ["jambi", "jambi", "Hari Nur Cahya Murni"]
15. bengkulu= ["bengkulu", "bengkulu", "rohidin
    mersyah"]
16. bangka_belitung = ["bangka belitung", "belitung",
    "erzaldi rosman"]#15
17. kalimantan_barat = ["kalimantan barat",
    "kalbar","sutarmidji"]
18. kalimantan_timur = ["kalimantan timur", "kaltim",
    "isran noor"]#15
19. kalimantan_tengah = ["kalimantan tengah",
    "kalteng","sugianto sabran"]
20. kalimantan_selatan = ["kalimantan
    selatan","kalsel", "sahbirin noor"]
21. kalimantan_utara = ["kalimantan utara", "kalut",
    "zainal arifin"]
22. sulawesi_utara = ["sulawesi utara", "sulut", "olly
    dondokambey"]
23. gorontalo = ["gorontalo", "gorontalo", "rusli
    habibie"] #20
24.
```

```
25. sulawesi_tengah = ["sulawesi tengah", "sulteng",
    "longki djanggola"]
26. sulawesi_barat = ["sulawesi barat", "sulbar", "ali
    baal masdar"]
27. sulawesi_selatan = ["sulawesi selatan", "sulsel",
    "andi sudirman sulaiman"]
28. sulawesi_tenggara = ["sulawesi tenggara", "sultra",
    "ali mazi"]
29. bali = ["bali", "bali" "i wayan koster"] #25
30. nusa_tenggara_barat = ["nusa tenggara barat",
    "ntb", "zulkieflimansyah"]
31. nusa_tenggara_timur = ["nusa tenggara timur",
    "ntt", "viktor laiskodat"]
32. maluku = ["maluku", "maluku", "murad ismail"]
33. maluku_utara = ["maluku utara", "malut", "abdul
    ghani kasuba"]
34. papua = ["papua", "papua", "lukas enembe"]
35. papua_barat = ["papua barat", "pabbar", "dominggus
    mandacan"]
36. nasional = ["psbb","vaksin","pembatasan sosial
    berskala besar","jokowi","joko
    widodo","indonesia","menteri","menteri
    kesehatan","menkes","kemenkes","kementerian
    kesehatan","terawan","budi gunadi sadikin","achmad
    yurianto","pilkada","satgas","mudik","puasa","rapid
    tes","rapid antigen","swab tes","tes swab","tes
    rapid","update covid","update
    corona","bpjs","tenaga medis","tenaga
    kesehatan","nakes"]
37.
38. #kata_kunci kota
39. jawa_timur.extend(["surabaya","sidoarjo","gresik",
    "malang","bangkalan","banyuwangi","blitar","bojoneg
    oro","bondowoso","jember","jombang","kediri","lamon
    gan","lumajang","madiun","magetan","mojokerto","nga
    njuk","ngawi","pacitan","pamekasan","pasuruan","pon
    orogo","probolinggo","sampang","sidoarjo","situbond
    o","sumenep","trenggalek","tuban","tulungagung","ba
    tu","kediri"])
40. jawa_tengah.extend(["banjarnegara","banyumas","bata
    ng","blora","boyolali","brebes","cilacap","demak","
    grobogan","jepara","karanganyar","kebumen","kendai"
    ,"klaten","kudus","magelang","pati","pekalongan","p
    emalang","purbalingga","purworejo","rembang","semar
    ang","sragen","sukoharjo","tegal","temanggung","won
    ogiri","wonosobo","magelang","pekalongan","salatiga
    ","surakarta"])
```

```
41. dki_jakarta.extend(["jakarta utara","jakarta
    barat","jakarta pusat","jakarta selatan","jakarta
    timur"])
42. jawa_barat.extend(["bandung
    barat","bandung","bekasi","bogor","ciamis","cianjur
    ","cirebon","garut","indramayu","karawang","kuninga
    n","majalengka","pangandaran","purwakarta","minahas
    a","subang","sukabumi","sumedang","tasikmalaya","ba
    njar","bekasi","cimahi","depok"])
43. di_yogyakarta.extend(["bantul","gunung
    kidul","kulon progo","sleman","yogyakarta"])
44. banten.extend(["lebak","pandeglang","serang","cileg
    on","tangerang selatan","tangerang","tangsel"])
45. aceh.extend(["aceh barat daya","aceh barat","aceh
    besar","aceh jaya","aceh selatan","aceh
    singkil","aceh tamiang","aceh tengah","aceh
    tenggara","aceh timur","aceh utara","bener
    meriah","bireuren","gayo lues","nagan raya","pidie
    jaya","pidie","simeulue","banda
    aceh","langsa","lhokseumawe","sabang","subulussalam
    "])
46. sumatera_barat.extend(["agam","dharmasraya","mentaw
    ai","lima puluh kota","padang pariaman","pasaman
    barat","pasaman","pasaman barat","pesisir
    selatan","sijunjung","solok selatan","solok","tanah
    datar","bukit tinggi","padang
    panjang","padang","pariaman","payakumbuh","sawahlun
    to","solok"])
47. sumatera_utara.extend(["asahan","batu
    bara","dairi","deli serdang","humbang
    hasundutan","karo","labuhanbatu
    selatan","labuhanbatu
    utara","labuhanbatu","langkat","mandailing
    natal","niat barat","nias selatan","nias
    utara","nias","padang lawas utara","padang
    lawas","pakpak bharat","samosir","serdang
    berdagai","simalungun","tapanuli selatan","tapanuli
    tengah","tapanuli
    utara","toba","binjai","gunungsitoli","medan","pada
    ngsidempuan","pematangsiantar","sibolga","tanjungba
    lai","tebing tinggi"])
48. sumatera_selatan.extend(["banyuasin","empat
    lawang","lahat","muara enim","musi banyuasin","musi
    rawas utara","musi rawas","ogan ilir","ogan
    komering ilir","ogan komering ulu selatan","ogan
    komering ulu timur","ogan komering ulu","penukal
    abab lematang ilir","lubuklinggau","pagar
    alam","palembang","prabumulih"])
```

```
49. riau.extend(["bengkalis","indragiri
    hilir","indragiri hulu","kampar","kepulauan
    meranti","kuantan singingi","pelalawan","rokan
    hilir","rokan hulu","siak","dumai","pekanbaru"])
50. kep_riau.extend(["bintan","karimun","anambas","ling
    ga","natuna","batam","tanjungpinang"])
51. lampung.extend(["lampung barat","lampung
    selatan","lampung tengah","lampung timur","lampung
    utara","mesuji","pesawaran","pesisir
    barat","pringsewu","tanggamus","tulang bawang
    barat","tulang bawang","way kanan","bandar
    lampung","metro lampung"])# metro
52. jambi.extend(["batanghari","bungo","kerinci","meran
    gin","muaro jambi","sarolangun","tanjung jabung
    barat","tanjung jabung
    timur","jambi","sungaipenuh"])
53. bengkulu.extend(["bengkulu selatan","bengkulu
    tengah","bengkulu
    utara","kaur","kepahiang","lebong","mukomuko","reja
    ng lebong","seluma","bengkulu"])
54. bangka_belitung.extend(["bangka barat","bangka
    tengah","bangka
    selatan","bangka","belitung","belitung
    timur","pangkalpinang"])
55. kalimantan_barat.extend(["bengkayang","kapuas
    hulu","kayong utara","ketapang","kubu
    raya","landak","melawi","mempawah","sambas","sangga
    u","sekadau","sintang","potianak","singkawang"])
56. kalimantan_timur.extend(["berau","kutai
    barat","kutai kartanegara","kutai timur","mahakam
    ulu","paser","penajam paser
    utara","balikpapan","bontang","samarinda"])
57. kalimantan_tengah.extend(["barito selatan","barito
    timur","barito utara","gunung
    mas","kapuas","katingan","kotawaringin
    barat","kotawaringin timur","lamandau","murung
    raya","pulang
    pisau","sukamara","seruyan","palangkaraya"])
58. kalimantan_selatan.extend(["balangan","banjar","bar
    ito kuala","hulu sungai selatan","hulu sungai
    tengah","hulu sungai
    utara","kotabaru","tabalong","tanah bumbu","tanah
    laut","tapin","banjarbaru","banjarmasin"])
59. kalimantan_utara.extend(["bulungan","malinau","nunu
    kan","tana tidung","tarakan"])
60. sulawesi_utara.extend(["bolaang mongondow
    selatan","bolaang mongondow utara","bolaang
    mongondow timur","bolaang
    mongondow","sangihe","siau tagulandang
```

```
        biaro","talaud","minahasa selatan","minahasa
        tenggara","minahasa
        utara","minahasa","bitung","kotamobagu","manado","t
        omohon"])
    61. gorontalo.extend(["boalemo","bone
        bolango","gorontalo","gorontalo utara","pohutawo"])
    62. sulawesi_tengah.extend(["banggai
        kepulauan","banggai
        laut","banggai","buol","donggala","morowali
        utara","morowali","parigi
        moutong","poso","sigi","tojo una-
        una","tolitoli","palu"])
    63. sulawesi_barat.extend(["majene","mamasa","mamuju
        tengah","mamuju","pasangkayu","polewali mandar"])
    64. sulawesi_selatan.extend(["bantaeng","barru","bone",
        "bulukumba","enrekang","gowa","jeneponto","selayar"
        ,"luwu timur","luwu
        utara","luwu","majonange","pangkep","pinrang","side
        nreng rappang","sinjai","soppeng","takalar","tana
        toraja","toraja
        utara","wajo","makassar","palopo","parepare"])
    65. sulawesi_tenggara.extend(["bombana","buton
        selatan","buton tengah","buton
        utara","buton","kolaka timur","kolaka
        utara","kolaka","konawe","konawe kepulauan","konawe
        selatan","konawe utara","muna
        barat","muna","wakatobi","bau-bau","kendari"])
    66. bali.extend(["badung","bangli","buleleng","gianyar"
        ,"jembrana","karangasem","klukung","tabanan","denpa
        sar"])
    67. nusa_tenggara_barat.extend(["bima","dompu","lombok
        barat","lombok timur","lombok utara","lombok
        tengah","lombok","sumbawa barat","sumbawa
        barat","bima","mataram"])
    68. nusa_tenggara_timur.extend(["alor","belu","ende","f
        lores timur","kupang","lembata","malaka","manggarai
        barat","manggarai
        timur","manggarai","nagekeo","ngada","rote
        ndao","sabu raijua","sikka","sumba barat
        daya","sumba barat","sumba timur","sumba
        tengah","timor tengah selatan","timor tengah
        utara","kupang"])
    69. maluku.extend(["buru selatan","kepulauan
        aru","maluku barat daya","maluku tengah","maluku
        tenggara","kepulauan tanimbar","seram bagian
        barat","seram bagian timur","ambon","tual"])
    70. maluku_utara.extend(["halmahera barat","halmahera
        tengah","halmahera timur","halmahera
        selatan","halmahera utara","kepulauan sula","pulau
```

```
    morotai","pulau taliabu","ternate","tidore
    kepulauan"])
71. papua.extend(["asmat","biak numfor","boven
    digoel","deiyai","dogiyai","intan
    jaya","jayapura","jayawijaya","keerom","kepulauan
    yapen","lanny jaya","mamberamo raya","mamberamo
    tengah","mappi","merauke","mimika","nabire","nduga"
    ,"paniai","pegunungan bintang","puncak
    jaya","puncak","sarmi","supiori","tolikara","warope
    n","yahukimo","yalimo"])
72. papua_barat.extend(["fakfak","kaimana","manokwari
    selatan","manokwari
    selatan","manokwari","maybrat","pegunungan
    arfak","raja ampat","sorong
    selatan","sorong","tambrauw","teluk bintuni","teluk
    wondama"])
```

Kode Sumber 7.1Klasifikasi Teks Menggunakan SVM

2. Support Vector Machine (SVM)

```
1.  path_df = "C:/Users/asus-pc/Documents/PBA/Tugas
    Akhir/Untitled Folder/Pickles_title/df.pickle"
2.  with open(path_df, 'rb') as data:
3.      df = pickle.load(data)
4.
5.  path_features_train = "C:/Users/asus-
    pc/Documents/PBA/Tugas Akhir/Untitled
    Folder/Pickles_title/features_train.pickle"
6.  with open(path_features_train, 'rb') as data:
7.      features_train = pickle.load(data)
8.
9.  path_labels_train = "C:/Users/asus-
    pc/Documents/PBA/Tugas Akhir/Untitled
    Folder/Pickles_title/labels_train.pickle"
10. with open(path_labels_train, 'rb') as data:
11.     labels_train = pickle.load(data)
12.
13. C = [.0001, .001, .01]
14. gamma = [.0001, .001, .01, .1, 1, 10, 100]
15. degree = [1, 2, 3, 4, 5]
16. kernel = ['linear', 'rbf', 'poly']
17. probability = [True]
18.
```

```
19. random_grid = {'C': C,
20.                'kernel': kernel,
21.                'gamma': gamma,
22.                'degree': degree,
23.                'probability': probability
24.               }
25. svc = svm.SVC(random_state=8)
26.
27. random_search = RandomizedSearchCV(estimator=svc,
28.
    param_distributions=random_grid,
29.                        n_iter=50,
30.                        scoring='accuracy',
31.                        cv=3,
32.                        verbose=1,
33.                        random_state=8)
34.
35. # Fit the random search model
36. random_search.fit(features_train, labels_train)
37. # Create the parameter grid based on the results of
    random search
38. C = [.0001, .001, .01, .1]
39. degree = [3, 4, 5]
40. gamma = [1, 10, 100]
41. probability = [True]
42.
43. param_grid = [
44.   {'C': C, 'kernel':['linear'],
    'probability':probability},
45.   {'C': C, 'kernel':['poly'], 'degree':degree,
    'probability':probability},
46.   {'C': C, 'kernel':['rbf'], 'gamma':gamma,
    'probability':probability}
47. ]
48.
49. svc = svm.SVC(random_state=8,gamma='scale')
50.
51. cv_sets = ShuffleSplit(n_splits = 3, test_size =
    .33, random_state = 8)
52.
53. grid_search = GridSearchCV(estimator=svc,
54.                            param_grid=param_grid,
55.                            scoring='accuracy',
56.                            cv=cv_sets,
57.                            verbose=1)
58.
59. grid_search.fit(features_train, labels_train)
60.
61. best_svc = grid_search.best_estimator
```

```
best_svc.fit(features_train, labels_train)

print(accuracy_score(labels_train,
best_svc.predict(features_train)))
```

Kode Sumber 7.2 Klasifikasi Teks Menggunakan SVM

3. K-Nearest Neighbors

```
path_df = "C:/Users/asus-pc/Documents/PBA/Tugas
Akhir/Untitled Folder/Pickles_title/df.pickle"
with open(path_df, 'rb') as data:
    df = pickle.load(data)

path_features_train = "C:/Users/asus-
pc/Documents/PBA/Tugas Akhir/Untitled
Folder/Pickles_title/features_train.pickle"
with open(path_features_train, 'rb') as data:
    features_train = pickle.load(data)

path_labels_train = "C:/Users/asus-
pc/Documents/PBA/Tugas Akhir/Untitled
Folder/Pickles_title/labels_train.pickle"
with open(path_labels_train, 'rb') as data:
    labels_train = pickle.load(data)

n_neighbors = [int(x) for x in np.linspace(start =
1, stop = 500, num = 100)]

param_grid = {'n_neighbors': n_neighbors}

knnc = KNeighborsClassifier()

cv_sets = ShuffleSplit(n_splits = 3, test_size =
.33, random_state = 8)
grid_search = GridSearchCV(estimator=knnc,
                           param_grid=param_grid,
                           scoring='accuracy',
                           cv=cv_sets,
                           verbose=1)

grid_search.fit(features_train, labels_train)
```

```
27. n_neighbors = [51,52,53,54,55,56,57,58,59,60,61]
28. param_grid = {'n_neighbors': n_neighbors}
29.
30. knnc = KNeighborsClassifier()
31. cv_sets = ShuffleSplit(n_splits = 3, test_size =
    .33, random_state = 8)
32.
33. grid_search = GridSearchCV(estimator=knnc,
34.                            param_grid=param_grid,
35.                            scoring='accuracy',
36.                            cv=cv_sets,
37.                            verbose=1)
38.
39. grid_search.fit(features_train, labels_train)
40.
41. best_knnc = grid_search.best_estimator_
42. best_knnc.fit(features_train, labels_train)
43. print(accuracy_score(labels_train,
    best_knnc.predict(features_train)))
```

Kode Sumber 7.3 Klasifikasi Teks Menggunakan KNN

4. Multinomial Naïve Bayes

```
1.  path_df = "C:/Users/asus-pc/Documents/PBA/Tugas
    Akhir/Untitled Folder/Pickles_title/df.pickle"
2.  with open(path_df, 'rb') as data:
3.      df = pickle.load(data)
4.
5.  path_features_train = "C:/Users/asus-
    pc/Documents/PBA/Tugas Akhir/Untitled
    Folder/Pickles_title/features_train.pickle"
6.  with open(path_features_train, 'rb') as data:
7.      features_train = pickle.load(data)
8.
9.  path_labels_train = "C:/Users/asus-
    pc/Documents/PBA/Tugas Akhir/Untitled
    Folder/Pickles_title/labels_train.pickle"
10. with open(path_labels_train, 'rb') as data:
11.     labels_train = pickle.load(data)
12.
13.
```

```
14. mnbc = MultinomialNB()
15. mnbc.fit(features_train, labels_train)
16. print(accuracy_score(labels_train,
    mnbc.predict(features_train)))
```

[*Halaman ini sengaja dikosongkan*]

BIODATA PENULIS

Muhammad Naufal Refadi, lahir di Gresik tanggal 23 April 1999. Penulis merupakan anak terakhir dari dua bersaudara. Penulis telah menempuh pendidikan formal di PG/TK Tadika Puri (2001-2005), SD Muhammadiyah GKB Gresik (2005-2011), SMP Negeri 1 Gresik (2011-2014), SMA Semesta Bilingual Boarding School Semarang (2014-2017). Penulis melanjutkan studi dengan berkuliah pada program sarjana (S1) di Departemen Teknik Informatika ITS. Selama kuliah di Teknik Informatika ITS, penulis mengambil bidang minat Algoritma Pemrograman (AP). Selain itu penulis aktif mengikuti organisasi, yaitu sebagai staff dan staff ahli Himpunan Mahasiswa Teknik Computer-Informatika pada tahun 2019-2020. Selain aktif mengikuti organisasi. Penulis juga aktif dalam kegiatan kepanitiaan *event* nasional dan jurusan, yaitu sebagai panitia *National Programming Competitio*n (NPC) pada tahun 2018-2019 ketua Hackathon Informatika ITS pada tahun 2019, dan ketua Quadrathlo Informatika ITS pada tahun 2020. Penulis dapat dihubungi melalui surel di naufal.refadi@yahoo.co.id.