

# SKRIPSI

## ANALISIS SENTIMEN BERBASIS ASPEK PADA ULASAN TEMPAT WISATA DIY



Disusun Oleh:

N a m a : Auliya Khanza Qorita

NIM : 18523214

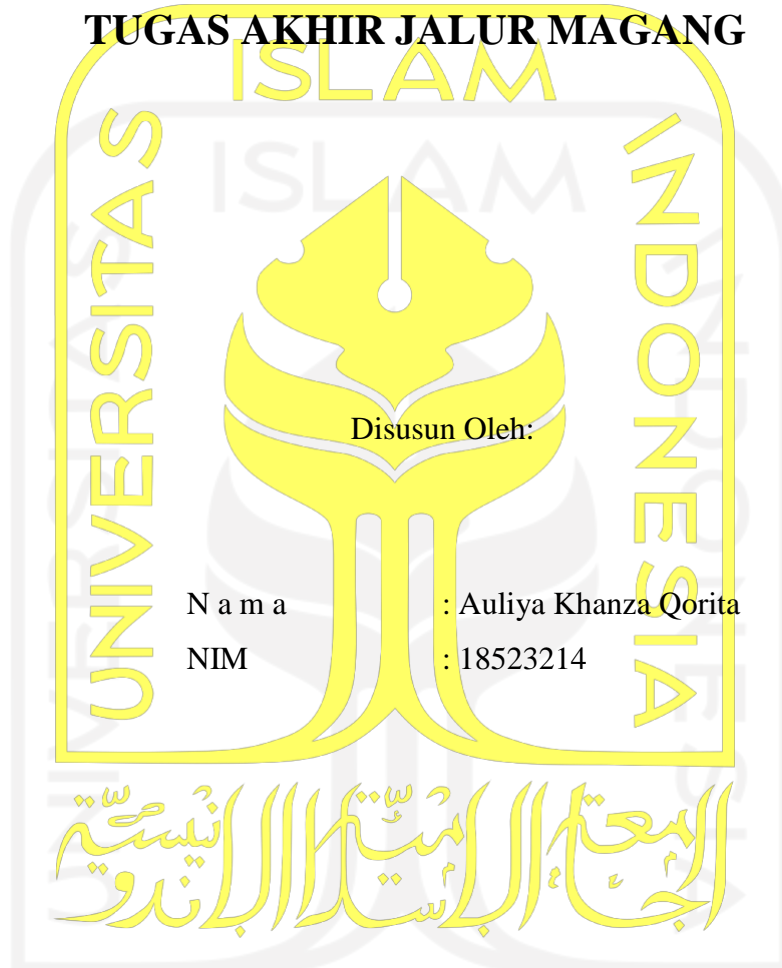
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA**

**2022**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**ANALISIS SENTIMEN BERBASIS ASPEK PADA ULASAN  
TEMPAT WISATA DIY**

**TUGAS AKHIR JALUR MAGANG**



Disusun Oleh:

N a m a : Auliya Khanza Qorita  
NIM : 18523214

Yogyakarta, 27 Mei 2022

Pembimbing,

Signer ID: YWYUEVEMF8...

( Fayruz Rahma, S.T., M.Eng. )

**HALAMAN PENGESAHAN DOSEN PENGUJI**

**ANALISIS SENTIMEN BERBASIS ASPEK PADA ULASAN**

**TEMPAT WISATA DIY**

**TUGAS AKHIR JALUR MAGANG**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 27 Mei 2022

Tim Penguji

**Ketua Penguji**

Fayruz Rahma, S.T., M.Eng.



**Anggota 1**

Sheila Nurul Huda, S.Kom, M.Cs.



**Anggota 2**

Septia Rani, S.T., M.Cs.



Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc. )

**HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertanda tangan di bawah ini:

Nama : Auliya Khanza Qorita  
NIM : 18523214

Tugas akhir dengan judul:

**ANALISIS SENTIMEN BERBASIS ASPEK PADA ULASAN  
TEMPAT WISATA DIY**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 25 April 2022



( Auliya Khanza Qorita )

## HALAMAN PERSEMBAHAN

Laporan akhir jalur magang ini dipersembahkan untuk kedua orang tua saya yang telah memberikan segalanya untuk saya dari kecil sampai sekarang. Kemudian, laporan akhir ini juga dipersembahkan kepada kakak dan sahabat saya yang telah memberikan dukungan dalam pengerjaan laporan akhir ini. Terakhir, saya persembahkan laporan akhir ini untuk siapa saja yang nantinya akan membutuhkan laporan akhir ini untuk dijadikan bahan referensi.



**HALAMAN MOTO**

“... dan aku menyerahkan urusanku kepada Allah. Sungguh, Allah Maha Melihat akan hamba-hamba-Nya.”

(Qs. Al-Mu'min: 44)

*“Believe in yourself, your abilities and your own potential. Never allow self-doubt to hold you captive. You are worth everything you dream of and hope for.”*

(Roy Bennet)



## KATA PENGANTAR

*Assalamu'alaikum Wr. Wb.*

Alhamdulillah penulis ucapkan puji syukur serta nikmat kepada Allah SWT yang sudah limpahkan rahmat serta hidayah-Nya sehingga bisa selesaikan Laporan Tugas Akhir ini dengan baik.

Salawat serta salam tidak lupa penulis ucapkan pada junjungan kita Nabi Muhammad SAW, karena telah membimbing kita dari zaman jahiliyah (kebodohan) menuju zaman yang terang benderang. Laporan Tugas Akhir ini dibuat guna jadi bukti pelaksanaan kegiatan magang di Dinas Komunikasi dan Informatika DIY dan untuk memenuhi salah satu syarat kelulusan untuk jalur magang Jurusan “Informatika Universitas Islam Indonesia”.

Pada pembuatan Laporan Tugas Akhir ini, penulis menyadari dapatkan bimbingan, bantuan, serta dorongan dari banyak pihak. Untuk itu, penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua penulis yang telah memberikan do'a dan dukungannya selama kegiatan magang dan pembuatan laporan ini berlangsung.
2. Dr. Raden Teduh Dirgahayu, S.T., M. Sc., selalu Ketua Program Studi Informatika – Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Fayruz Rahma, S.T., M.Eng., selaku dosen pembimbing yang bersedia untuk membantu dan membimbing penulis dalam penyusunan laporan akhir ini.
4. Bapak dan Ibu dosen Jurusan Informatika Universitas Islam Indonesia yang telah memberikan ilmu, bimbingan, dan arahan yang bermanfaat bagi penulis.
5. Dinas Komunikasi dan Informatika DIY, beserta seluruh pimpinan dan staf yang telah memberi kesempatan untuk penulis melaksanakan magang dan membantu penulis dalam menyelesaikan laporan akhir ini.
6. Bapak Novan Hartadi, sebagai *Project Manager* Jogja Center dan seluruh pihak yang terlibat dalam *project* Jogja Center.
7. Teman-teman Informatika Universitas Islam Indonesia angkatan 2018 yang telah memberikan dukungan dan bantuan selama masa perkuliahan.
8. Semua pihak yang penulis sayangi dan banggakan yang sudah berikan *support* untuk penulis yang tidak bisa disebutkan satu-persatu.

Semoga seluruh kebaikan yang sudah diberi dapatkan balasan yang lebih baik lagi. Selain itu, penulis memohon maaf bila masih ada kesalahan pada laporan akhir ini. Penulis menerima kritik serta saran untuk menjadi perbaikan ke depannya. Semoga laporan akhir ini berguna bagi pembaca.

*Wassalamu'alaikum, Wr. Wb.*

Yogyakarta, 13 Desember 2021



( Auliya Khanza Qorita )





## SARI

Seiring berkembangnya teknologi, informasi-informasi mengenai situasi dan kondisi di tempat wisata dapat diakses melalui beberapa situs yang menyediakan ulasan mengenai tempat wisata. Ulasan yang telah ditulis para pengunjung tempat wisata dapat dijadikan bahan evaluasi oleh pemerintah daerah setempat untuk mengembangkan tempat wisata itu jadi lebih baik. Maka, untuk memudahkan pemerintah daerah dalam melihat ulasan tempat wisata, Dinas Komunikasi dan Informatika DIY mengembangkan sebuah situs web yang diberi nama Jogja Center. Dalam situs tersebut, terdapat dasbor Sentimen Tempat Wisata Menurut Data *Online Travel*. Dasbor tersebut akan menampilkan mayoritas kelas sentimen (positif, netral, dan negatif) dari ulasan yang telah diberikan oleh pengunjung tempat wisata. Untuk mengetahui mayoritas kelas sentimen perlu dibuat sebuah model yang dapat melakukan analisis sentimen berdasarkan aspek. Dalam pembuatan model tersebut, digunakan dua metode yaitu Multinomial Naïve Bayes dan Support Vector Machine. Untuk mengetahui apakah model yang dibuat sudah cukup baik atau belum dapat dengan melihat akurasi dan *f1-score* yang dihasilkan dari kedua metode yang digunakan. Berdasarkan hasil pengujian didapatkan bahwa akurasi yang dihasilkan oleh algoritma Support Vector Machine lebih baik daripada algoritma Multinomial Naïve Bayes. Algoritma SVM menghasilkan nilai 66.49% pada aspek attraction, 87.27% pada aspek accessible, 69.35% pada aspek amenities, dan 90.39% pada aspek ancillary services. Akan tetapi, nilai *f1-score* yang dihasilkan oleh algoritma SVM untuk sentimen *neutral* menunjukkan nilai yang tidak lebih baik daripada menggunakan metode Multinomial Naïve Bayes. Oleh karena itu, diberikan saran untuk mencari algoritma terbaik dalam melakukan klasifikasi teks dapat dengan mencoba menggunakan algoritma dan kombinasi lainnya sehingga dapat dijadikan sebagai pembanding dari model yang telah dibuat sebelumnya.

Kata kunci: Analisis Sentimen Berdasarkan Aspek, Ulasan, Tempat Wisata, Multinomial Naïve Bayes, Support Vector Machine.

## GLOSARIUM

<i>Classification Model</i>	salah satu model pada <i>machine learning</i> yang dapat membedakan dua atau lebih data berdasarkan kelas.
<i>Labelling</i>	proses pemberian label pada data berdasarkan kelas yang telah ditentukan.
<i>Modelling</i>	proses melatih algoritma <i>machine learning</i> untuk melakukan prediksi pada data.
<i>Multinomial Naïve Bayes</i>	salah satu metode <i>probabilistic reasoning</i> yang dapat mengklasifikasikan data ke dalam kelas tertentu.
<i>Preprocessing</i>	tahapan pengolahan data yang digunakan untuk mengubah data menjadi format tertentu yang berguna, efisien, dan konsisten.
SMOTE	metode <i>oversampling</i> yang populer dipakai guna menyeimbangkan dataset.
<i>Support Vector Machine</i>	metode pada <i>supervised learning</i> yang dapat digunakan untuk klasifikasi dan regresi.

## DAFTAR ISI

ANALISIS SENTIMEN BERBASIS ASPEK PADA ULASAN TEMPAT WISATA DIY	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
<b>BAB I PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang	1
1.2 Ruang Lingkup Magang	2
1.3 Tujuan	2
1.4 Manfaat	2
1.5 Sistematika Penulisan	3
<b>BAB II DASAR TEORI</b>	<b>4</b>
2.1 Data Labelling	4
2.2 Preprocessing	4
2.3 Analisis Sentimen	5
2.4 Analisis Sentimen Berdasarkan Aspek	5
2.5 Term Frequency – Inverse Document Frequency	5
2.6 Multinomial Naïve Bayes	6
2.7 Support Vector Machine	8
2.8 Confusion Matrix	11
<b>BAB III PELAKSANAAN MAGANG</b>	<b>13</b>
3.1 Analisis Sentimen pada Ulasan Tempat Wisata Berdasarkan Aspek	13
3.1.1 Data Labelling	14
3.1.2 Preprocessing	16
3.1.3 Klasifikasi	18
3.1.4 Evaluation	21
3.1.5 Testing	23
3.2 Analisis Sentimen Media Sosial Twitter	24
3.2.1 Data Labelling	24
3.2.2 Preprocessing	26
3.2.3 Klasifikasi	26
3.2.4 Evaluation	28
3.3 <i>Task</i> Kolaborasi	28
3.3.1 <i>List</i> Data Tempat Wisata di Yogyakarta	28
3.3.2 <i>List</i> Data Putusan Pengadilan di Yogyakarta	29
3.3.3 <i>Report</i> Korelasi Pergerakan Google Mobility dan Kenaikan Kasus COVID-19 di Yogyakarta	31
3.3.4 <i>Report</i> Vaksinasi di Yogyakarta	33
3.3.5 <i>Report</i> Sentimen Publik dan Hoaks Selama Pandemi	36
<b>BAB IV REFLEKSI PELAKSANAAN MAGANG</b>	<b>40</b>

4.1	Teknis	40
4.2	Non-teknis	41
4.2.1	Manfaat Magang	41
4.2.2	Kendala, Hambatan, dan Tantangan	41
4.2.3	Kontribusi Selama Magang	42
BAB V KESIMPULAN DAN SARAN		43
5.1	Kesimpulan	43
5.2	Saran	43
DAFTAR PUSTAKA		45
1.	LAMPIRAN	47



## DAFTAR TABEL

Tabel 2.1 Confusion Matrix .....	12
Tabel 3.1 Hasil Data <i>Labelling</i> .....	16
Tabel 3.2 <i>Case Folding</i> .....	16
Tabel 3.3 <i>Filtering</i> .....	16
Tabel 3.4 <i>Stopwords Removal</i> .....	17
Tabel 3.5 Hasil <i>F1-Score</i> SVM Kernel Linier .....	20
Tabel 3.6 Hasil <i>F1-Score</i> SVM Kernel RBF .....	20
Tabel 3.7 Hasil <i>F1-Score</i> SVM Kernel <i>Polynomial</i> .....	20
Tabel 3.8 Hasil Akurasi Model Multinomial Naïve Bayes .....	21
Tabel 3.9 Hasil <i>F1-Score</i> Multinomial Naive Bayes .....	22
Tabel 3.10 <i>Confusion Matrix</i> Aspek <i>Amenities</i> Multinomial Naive Bayes .....	22
Tabel 3.11 Hasil Akurasi Model Support Vector Machine.....	22
Tabel 3.12 Hasil <i>F1-Score</i> Model Support Vector Machine .....	23
Tabel 3.13 <i>Confusion Matrix</i> Aspek <i>Attraction</i> Support Vector Machine .....	23

## DAFTAR GAMBAR

Gambar 2.1 Ilustrasi SVM dalam mengelompokkan kelas +1 dan -1 serta menemukan <i>hyperplane</i> terbaik .....	9
Gambar 3.1 <i>Flowchart</i> Analisis Sentimen.....	14
Gambar 3.2 Tabel <i>Labelling</i> Tempat Wisata.....	15
Gambar 3.3 <i>Guide</i> Pelabelan Tempat Wisata.....	15
Gambar 3.4 Kode Program Sebelum <i>Pre-processing</i> .....	17
Gambar 3.5 Kode Program <i>Pre-processing</i> .....	18
Gambar 3.6 Distribusi Data Setelah Dilakukan SMOTE .....	19
Gambar 3.7 Kode Multinomial Naive Bayes.....	19
Gambar 3.8 Kode Support Vector Machine .....	21
Gambar 3.9 Kode Program dan Hasil Prediksi Model SVM.....	24
Gambar 3.10 <i>Labelling Rules</i> dan <i>Guidelines</i> .....	25
Gambar 3.11 Tabel <i>Labelling</i> Sentimen Negatif .....	25
Gambar 3.12 Persebaran Data Media Sosial.....	26
Gambar 3.13 Kode Program <i>Modelling</i> .....	27
Gambar 3.14 Hasil Evaluasi Sentimen Negatif .....	28
Gambar 3.15 <i>List</i> Tempat Wisata .....	29
Gambar 3.16 Kode Program <i>Task</i> Putusan Pengadilan .....	30
Gambar 3.17 <i>List</i> Putusan Pengadilan .....	31
Gambar 3.18 Data Google Mobility .....	32
Gambar 3.19 Data kasus COVID-19 .....	32
Gambar 3.20 Grafik Komparasi.....	33
Gambar 3.21 Data Vaksinasi Dosis Kedua.....	34
Gambar 3.22 Data Akumulasi Vaksin Dosis Kedua.....	35
Gambar 3.23 Grafik Vaksinasi .....	36
Gambar 3.24 Kode Program .....	37
Gambar 3.25 Kode Program <i>Timeseries</i> .....	37
Gambar 3.26 Sentimen Publik Bansos.....	38
Gambar 3.27 Sentimen Publik Malioboro .....	38
Gambar 3.28 Pemantauan Hoaks.....	39
Gambar 4.1 Dasbor <i>insight</i> data <i>platform online travel</i> .....	41

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dinas Komunikasi dan Informatika (Diskominfo) DIY ialah instansi pemerintahan yang bertugas untuk bantu Gubernur dalam melaksanakan urusan pemerintahan bidang komunikasi dan informatika dan bidang persandian. Lokasi Gedung Induk Diskominfo DIY bertempat di Jalan

Brigjen Katamso, Keparakan, Mergangsan, Kota Yogyakarta, Daerah Istimewa Yogyakarta. Pembentukan Diskominfo DIY ialah implementasi dari UU Nomor 23 Tahun 2014 tentang Pemerintah Daerah dan Peraturan Pemerintah Nomor 18 Tahun 2016 mengenai Perangkat Daerah.

Pada saat ini, Diskominfo DIY sedang mengembangkan sebuah web yang diberi nama Jogja Center. Jogja Center adalah inisiatif dalam pengembangan sistem analitis berbasis *Big Data*. Situs ini difokuskan pada pengembangan data *analytic* dan pendukung pengambilan keputusan, serta merujuk pada dimensi-dimensi Jogja *Smart Province* (JSP). Jogja Center ini dibuat sesuai dengan

visi dan misi yang dimiliki instansi. Visi Dinas Komunikasi dan Informatika DIY adalah “Terwujudnya Jogja *Cyber Province* dan Masyarakat Informasi Menuju Peradaban Baru Mendukung Keistimewaan DIY.” Kemudian, untuk misi dari Dinas Komunikasi dan Informatika DIY adalah “Mendukung peningkatan efisiensi dan efektivitas tata kelola pemerintah yang transparan dan akuntabel di Daerah Istimewa Yogyakarta dengan memanfaatkan teknologi informasi dan komunikasi yang terintegrasi secara optimal.”

Situs web ini terbagi jadi beberapa bagian yaitu: pariwisata, mobilitas, ekonomi, kesehatan, dan sosial. Pada bagian pariwisata, terdapat beberapa *insight* yang dapat dilihat salah satunya ialah *insight data platform online travel*. Perkembangan internet pada masa ini memiliki dampak dengan kemajuan yang pesat pada sektor pariwisata. Dengan internet, informasi mengenai tempat wisata dapat dengan mudah didapatkan melalui situs yang menyediakan ulasan tempat wisata. Ulasan tersebut dapat dijadikan sebagai bahan evaluasi pemerintah dalam pengembangan tempat wisata.

Pada *insight data platform online travel* ditampilkan persebaran data sentimen ulasan tempat wisata yang diolah menggunakan model *machine learning* yang telah dibuat. Model ini dibuat dikarenakan ulasan tempat wisata yang telah ditulis oleh pengunjung tempat wisata di *platform online travel* memiliki beberapa aspek dan polaritas yang berbeda. Hal itu dapat membuat kesulitan dalam menentukan polaritas yang ada pada ulasan tempat wisata tersebut.

Teknik yang digunakan dalam membuat model tersebut adalah *Aspect Based Sentiment Analysis* (ABSA). Teknik ABSA ini mengategorikan sentimen berdasarkan aspek menjadi beberapa kelas yaitu positif, negatif, dan netral. Metode yang dipakai ialah Multinomial Naïve Bayes dan Support Vector Machine. Dari kedua metode itu akan dilihat metode mana yang memiliki performa lebih baik dalam melakukan klasifikasi. Pada model ini digunakan 4 (empat) kategori aspek yaitu *attraction*, *accessible*, *amenities*, dan *ancillary services*. Adanya *insight* data *platform online travel* diharapkan dapat memudahkan Pemerintah DIY dalam melakukan evaluasi untuk meningkatkan kualitas tempat-tempat wisata di DIY.

## 1.2 Ruang Lingkup Magang

Pelaksanaan magang di Dinas Komunikasi dan Informatika DIY selama kurun waktu kurang lebih 6 (enam) bulan, dari bulan Maret 2021 sampai September 2021. Terdapat 6 (enam) unit kerja/bidang di Dinas Komunikasi dan Informatika DIY. Penulis melakukan magang di Bidang Keamanan Informasi dan Persandian. Pada bidang tersebut, penulis masuk ke tim Jogja Center. Penulis bertanggung jawab sebagai *Data Analyst* pada kegiatan magang ini.

Aktivitas yang dilakukan selama magang adalah mengolah data yang diberikan oleh *data engineer*, membuat visualisasi dari data yang telah diolah, lalu data-data tersebut akan disiapkan untuk dibuat model analisis sentimen yang digunakan untuk kebutuhan web Jogja Center. Adapun aktivitas lain yang dilakukan bersama dengan peserta magang yang lain adalah membuat *list* data tempat wisata yang terdapat di Yogyakarta, *list* data putusan pengadilan di Yogyakarta, *report* korelasi pergerakan antara data di Google Mobility dengan kenaikan kasus COVID-19 di Yogyakarta, *report* vaksinasi di Yogyakarta, dan *report* sentimen publik dan hoaks selama masa pandemi.

## 1.3 Tujuan

Tujuan dari pembuatan model analisis sentimen berdasarkan data *platform online travel* yang digunakan di web Jogja Center adalah untuk memudahkan Pemerintah DIY dalam mendapatkan umpan balik dari wisatawan mengenai tempat wisata yang dikelola.

## 1.4 Manfaat

Manfaat yang didapat dari pembuatan model analisis sentimen pada web Jogja Center adalah Pemerintah DIY dapat melakukan evaluasi mengenai tempat wisata yang dikelola sehingga dapat meningkatkan kualitas tempat wisata tersebut.



## **1.5 Sistematika Penulisan**

Untuk dapat mempermudah pembaca memahami laporan akhir ini, disusunlah sistematika penulisan. Sistematika penulisan dibuat:

SARI

BAB I PENDAHULUAN

Bab ini berisi latar belakang, ruang lingkup magang, tujuan, manfaat, dan sistematika penulisan.

BAB II DASAR TEORI

Bab ini berisi teori-teori yang mendukung proses pembuatan laporan.

BAB III PELAKSANAAN MAGANG

Bab ini berisi pelaksanaan dari kegiatan dan tugas yang telah diberi selama magang berlangsung.

BAB IV REFLEKSI PELAKSANAAN MAGANG

Bab ini berisi uraian hasil dan manfaat yang didapatkan penulis sesudah kegiatan magang.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran yang didapatkan dari kegiatan yang sudah dilakukan selama kegiatan magang berlangsung.

DAFTAR PUSTAKA

LAMPIRAN

## BAB II DASAR TEORI

Pada proses pengerjaan analisis sentimen berdasar aspek pada ulasan tempat wisata di Yogyakarta, terdapat beberapa tahap yang dilakukan. Berikut adalah dasar teori mengenai tahapan yang akan dijalankan di proses pengerjaan:

### 2.1 Data Labelling

Proses data *labelling* ini adalah proses mengidentifikasi data mentah lalu ditambahkan satu/lebih label yang informatif dan memiliki makna untuk memberikan konteks sehingga model *machine learning* dapat mempelajarinya. Kegiatan ini diperlukan untuk banyak kasus penggunaan seperti *computer vision*, *speech recognition*, dan pengolahan bahasa alami.

### 2.2 Preprocessing

*Preprocessing* adalah tahapan awal saat melakukan klasifikasi teks. Tahap *preprocessing* umumnya digunakan untuk mengurangi kesalahan pada data dalam data *raw* sebelum dilakukan analisis (Tong et al., 2011). Data yang digunakan biasanya diambil dari internet. Data-data *raw* tersebut umumnya tidak lengkap, berantakan, dan tidak konsisten. Tahap *preprocessing* data akan membuat data menjadi lebih rapi. Oleh karena itu, *preprocessing* merupakan langkah penting dalam proses penemuan pengetahuan, dikarenakan data yang berkualitas menghasilkan keputusan yang berkualitas (Kumar & Chadha, 2012).

Pada penelitian *Data Preprocessing for Supervised Learning* (Kotsiantis et al., 2007), dijelaskan bahwa terdapat beberapa faktor mempengaruhi keberhasilan *machine learning*. Faktor utama yang memengaruhi adalah representasi dan kualitas yang dimiliki data. Jika terdapat banyak data yang tidak relevan, memiliki banyak *noise*, redundansi data, dan data yang tidak handal maka untuk penemuan pengetahuan selama proses *training* lebih sulit. *Preprocessing* data membantu dalam melakukan data *cleaning*, data *reduction*, dan data *discretization*. Beberapa metode yang dilakukan pada saat *preprocessing* data dapat meningkatkan akurasi dari model yang menggunakan Naïve Bayes dan mengurangi waktu pemrosesan (Chandrasekar & Qian, 2016).

### 2.3 Analisis Sentimen

Analisis sentimen (*opinion mining*) merupakan proses untuk memahami dan mengolah data tekstual secara otomatis untuk menemukan informasi sentimen yang terdapat dalam suatu kalimat opini (Liesnawan, 2019). Kecenderungan suatu opini terhadap suatu masalah atau objek dapat dilihat dengan menggunakan analisis sentimen (Liesnawan, 2019). Secara umum, analisis sentimen terbagi menjadi 3 (tiga) level, yaitu level dokumen, level kalimat, serta level aspek dan entitas.

### 2.4 Analisis Sentimen Berdasarkan Aspek

Analisis sentimen berdasarkan aspek merupakan salah satu domain dalam *opinion mining* yang bertujuan untuk mendeteksi polaritas suatu data tekstual berdasarkan aspek tertentu (Septianti, 2020). Analisis sentimen berdasarkan aspek adalah proses untuk mendapatkan informasi suatu sentimen dari sudut pandang tertentu mengenai aspek tersebut (Liesnawan, 2019). Proses yang dilakukan pada analisis sentimen berdasarkan aspek adalah menentukan sentimen (positif, negatif, netral) berdasarkan aspek tertentu dalam suatu kalimat (Siregar, 2020).

Ada 4 (empat) komponen yang perlu ada pada tempat wisata yaitu *attraction*, *accessible*, *amenities*, dan *ancillary services* (Cooper et al., 1994). *Attraction* adalah segala sesuatu yang dapat menarik wisatawan untuk mengunjungi suatu tempat wisata. *Accessible* ialah kemudahan sarana transportasi dalam mencapai tempat wisata. *Amenities* adalah pelayanan dan fasilitas yang diperlukan oleh wisatawan saat berada di tempat wisata seperti penyediaan makanan dan minuman, *retailing*, fasilitas untuk pemenuhan kebutuhan akomodasi, dan layanan-layanan lainnya. *Ancillary services* adalah pelayanan tambahan yang disediakan oleh pemerintah daerah tempat wisata, kelompok, organisasi, atau pengelola tempat wisata.

### 2.5 Term Frequency – Inverse Document Frequency

TF-IDF adalah cara yang biasa dipakai guna melakukan pencarian informasi. TF-IDF memberikan bobot berbasis statistik, di mana tiap kata diberikan bobot. Selanjutnya, kalimat tersebut diurutkan berdasarkan bobotnya (Wahyunita, 2018). Metode ini sering digunakan dikarenakan mudah, efisien, serta memiliki hasil akurat (Robertson, 2004). Metode ini mengukur nilai dari “*Term-Frequency (TF)*” dengan “*Inverse Document Frequency (IDF)*” untuk tiap kata di tiap dokumen. Perhitungan bobot tiap token  $t$  di dokumen dapat dirumuskan:

$$W_{dt} = tf_{dt} * IDF_t \quad (2.1)$$

Keterangan:

$W_{dt}$  : bobot dari t (kata) dalam satu dokumen

$tf_{dt}$  : frekuensi kemunculan t (kata) dalam dokumen d

$IDF_t$  : *Inversed Document Frequency*

Nilai  $IDF_t$  didapatkan dari perhitungan sebagai berikut:

$$IDF_t = \log \left( \frac{N}{n_t} \right) \quad (2.2)$$

Keterangan:

$N$  : jumlah semua dokumen

$n_t$  : jumlah dokumen yang mengandung kata t

Setelah mengetahui bobot ( $W$ ) di masing-masing kata yang terdapat pada dokumen, selanjutnya akan dilakukan tahap *sorting* yaitu nilai bobot yang semakin besar, maka tingkat kesamaan dokumen pada kata kunci tersebut juga akan semakin besar, ataupun sebaliknya.

## 2.6 Multinomial Naïve Bayes

Metode Naïve Bayes ialah suatu metode yang efektif untuk digunakan dalam bidang klasifikasi teks, tetapi akurasi model menjadi lebih akurat apabila menggunakan *training sample set* yang banyak (Huang & Li, 2011). Metode Multinomial Naïve Bayes ini digunakan dalam melakukan klasifikasi teks dikarenakan kecepatan dan kesederhanaannya dalam melakukan klasifikasi (Goel et al., 2016). Metode ini mengikuti prinsip dari distribusi multinomial dalam melakukan pemrosesan teks (Farisi et al., 2019).

Metode Multinomial Naïve Bayes tidak hanya menghitung kata yang muncul tetapi juga jumlah kemunculan setiap kata yang ada pada dokumen (Witten et al., 2011). Metode ini mengasumsikan dokumen memiliki beberapa kejadian pada kata dengan panjang yang tidak tergantung dari kelas dokumen (Yanti, 2018). Menurut (Manning et al., 2008), perhitungan probabilitas sebuah dokumen  $d$  berada di kelas  $c$  menggunakan Multinomial Naïve Bayes dapat dirumuskan sebagai berikut:

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \quad (2.3)$$

Keterangan:

$P(t_k|c)$  merupakan *conditional probability* dari kata  $t_k$  yang terdapat dalam sebuah dokumen kelas  $c$ .

$P(c)$  adalah *prior probability* dari sebuah dokumen yang terdapat pada kelas  $c$ . Apabila kata dari suatu dokumen memberikan petunjuk yang kurang jelas untuk satu kelas dibandingkan kelas lainnya, maka akan dipilih satu kelas yang memiliki *prior probability* tertinggi.

$\langle t_1, t_2, \dots, t_{n_d} \rangle$  merupakan kumpulan *token* pada dokumen  $d$  yang merupakan bagian dari *vocabulary* yang digunakan saat melakukan klasifikasi dan  $n_d$  adalah jumlah *tokens* yang terdapat dalam  $d$ . Sebagai contoh,  $\langle t_1, t_2, \dots, t_{n_d} \rangle$  untuk satu kalimat pada dokumen “*Beijing and Taipei join the WTO*” menjadi  $\langle \text{Beijing}, \text{Taipei}, \text{join}, \text{WTO} \rangle$ , dengan  $n_d = 4$ , apabila diasumsikan kata *the* dan *and* adalah *stop words*.

Untuk menghitung nilai *prior probability*  $P_c$  diformulakan sebagai berikut:

$$P(c) = \frac{N_c}{N} \quad (2.4)$$

Keterangan:

$N_c$  : jumlah dari dokumen dalam kelas  $c$ .

$N$  : jumlah keseluruhan dokumen dari seluruh kelas.

Untuk menghitung nilai dari *conditional probability*  $P(t_k|c)$  dapat dilakukan menggunakan persamaan 2.5:

$$P(t_k|c) = \frac{T_{ct}}{\sum_{t^f \in V} T_{ct^f}}, \quad (2.5)$$

Keterangan:

$T_{ct}$  : jumlah kemunculan kata  $t$  dalam sebuah dokumen dalam kelas  $c$ .  
 $\sum_{t^f \in V} T_{ct^f}$  : jumlah total dari keseluruhan kata yang terdapat pada dokumen dalam kelas  $c$ .

Perhitungan nilai *prior probability*  $P_c$  dan *conditional probability*  $P(t_k|c)$  menggunakan *maximum likelihood estimate* (MLE). Masalah yang terjadi pada penggunaan MLE adalah terdapat nilai nol dari kombinasi kelas yang tidak terdapat dalam dokumen *training*. Sebagai contoh, apabila kata WTO pada dokumen *training* hanya terdapat pada dokumen China, maka prediksi untuk kelas-kelas lainnya, misal UK, akan bernilai 0 (nol).

$$P(WTO|UK) = 0 \quad (2.6)$$

Untuk menghilangkan nilai nol pada dokumen tersebut, dapat digunakan *Laplace smoothing* sebagai proses menambahkan nilai 1 (satu) untuk setiap nilai  $T_{ct}$  dari perhitungan *conditional probability*. Maka, formula untuk *conditional probability* menjadi seperti:

$$P(t_k|c) = \frac{T_{ct} + 1}{(\sum_{t^f \in V} T_{ct^f}) + B'} \quad (2.7)$$

Keterangan:

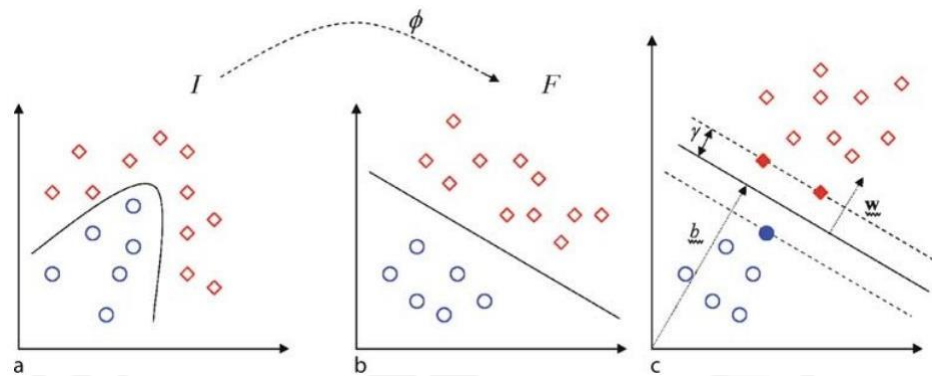
$B'$  : jumlah keseluruhan kata untuk dari seluruh kelas.

## 2.7 Support Vector Machine

Support Vector Machine adalah metode yang efektif untuk melakukan prediksi, baik itu dalam kasus klasifikasi maupun regresi (Cortes & Vapnik, 1995). SVM menggunakan hipotesis linier yang berfungsi dalam ruang dimensi tinggi berfitur lengkap. Pada ruang berdimensi tinggi akan dicari *hyperplane* terbaik yang dapat memisahkan 2 (dua) *classes* di ruang *input* (Fitriawan et al., 2013). SVM dapat digunakan untuk kasus klasifikasi linier ataupun *nonlinier*. Menurut (Santosa, 2007) dalam kasus klasifikasi secara linier, fungsi SVM di mana  $w$  sebagai vektor bobot (*weight vector*) dan  $b$  sebagai *bias* dapat dituliskan seperti persamaan 2.8:

$$f(x) = wx + b \quad (2.8)$$

Fungsi pada persamaan 2.8 digunakan untuk mengelompokkan nilai di atas 0 menjadi +1 dan di bawah 0 menjadi -1. Dalam metode ini, perlu ditemukan fungsi pemisah (*hyperplane*) terbaik untuk memisahkan dua kelas. Pada Gambar 2.1 ditunjukkan ilustrasi SVM untuk klasifikasi linier.



Gambar 2.1 Ilustrasi SVM dalam mengelompokkan kelas +1 dan -1 serta menemukan *hyperplane* terbaik

Sumber: Cristianini dan Ricci (2008)

Gambar 2.1 di grafik (a) dan (b) menunjukkan pemetaan fitur data menjadi kelas +1 dan -1 dipisahkan dengan garis pemisah (*discrimination boundaries*). Selanjutnya, grafik (c) menunjukkan garis *hyperplane* yang berada tepat di tengah dua *object* dari dua kelas. *Hyperplane* untuk kelas +1 dan -1 tersebut dapat dinotasikan sebagai berikut:

$$\begin{aligned} [(w \cdot x_i) + b] &\geq 1 \text{ untuk } y_i = +1 \\ [(w \cdot x_i) + b] &\leq -1 \text{ untuk } y_i = -1 \end{aligned} \quad (2.9)$$

Keterangan:

- $x_i$  : himpunan data *training*
- $i$  : 1, 2, ...,  $n$
- $y_i$  : label kelas dari  $x_i$

*Hyperplane* yang terletak di tengah antara 2 (dua) set *object* dari dua kelas adalah *hyperplane* terbaik. Untuk mendapatkan *hyperplane* terbaik dapat dengan cara memaksimalkan margin (Santosa, 2007). Margin dapat dihitung dengan mencari jarak antara kedua *hyperplane* pendukung dari 2 (dua) kelas yang dirumuskan sebagai berikut:

$$(wx_1 + b = 1) - (wx_2 + b = -1) \rightarrow w(x_1 - x_2) = 2 \quad (2.10)$$

Kemudian nilai maksimum margin dapat diformulakan menjadi persamaan 2.11:

$$\frac{w(x_1 - x_2)}{\|w\|} = \frac{2}{\|w\|} \quad (2.11)$$

Nilai *hyperplane* optimal ialah  $\max \frac{2}{\|w\|}$  atau ekuivalen dengan  $\min \frac{1}{2} \|w\|^2$ .

Secara matematis, formula masalah optimasi SVM untuk klasifikasi linier di dalam *primal space* adalah

$$\min \frac{1}{2} \|w\|^2 \quad (2.12)$$

Dengan syarat kendala  $y_i(w^T \cdot x_i + b) \geq 1$ ,  $i = 1, 2, 3, \dots, n$ .

Solusi untuk permasalahan optimasi tersebut dapat diselesaikan dengan mengubah formula 2.12 ke dalam formula *lagrange* seperti yang dituliskan pada persamaan 2.13:

$$L_{pri}(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i(wx_i + b) - 1] \quad (2.13)$$

Dengan nilai dari koefisien *lagrange*  $\alpha_i \geq 0$ . Untuk menyelesaikan permasalahan pada persamaan 2.13 dengan mencari *saddle point* dari  $L_{pri}$ . Fungsi  $L_{pri}$  harus diminimalkan terhadap variabel  $w$  serta  $b$  dan harus dimaksimalkan terhadap variabel  $\alpha$ . Untuk meminimalkan  $L_{pri}$  dengan mencari turunan pertama  $L_{pri}$  terhadap variabel  $w$  dan  $b$  lalu disamakan dengan 0. Proses perhitungan tersebut menghasilkan 2 (dua) kondisi optimalitas yaitu:

Kondisi 1.

$$\frac{\partial J(w, b, \alpha)}{\partial w} = 0 \quad (2.14)$$

Kondisi 2.

$$\frac{\partial J(w, b, \alpha)}{\partial b} = 0 \quad (2.15)$$

Penerapan dari kedua kondisi tersebut akan menghasilkan persamaan 2.16:



$$w = \sum_{i=1}^n \alpha_i y_i x_i \text{ dan } \sum_{i=1}^n \alpha_i y_i = 0 \quad (2.16)$$

Di mana  $n$  adalah jumlah data *support*.

Dikarenakan nilai dari  $w$  seringkali tak terhingga tetapi nilai  $\alpha_i$  terhingga, formula  $L_{pri}$  (*primal problem*) diubah menjadi  $L_D$  (*dual problem*). Dengan cara melakukan substitusi persamaan 2.16 ke persamaan 2.13 diperoleh *dual problem* seperti yang dituliskan pada persamaan 2.17:

$$L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (2.17)$$

Sehingga permasalahan mencari *hyperplane* terbaik dapat diformulakan pada persamaan 2.18:

$$\max L_D = \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (2.18)$$

Di mana  $\sum_{j=1}^n \alpha_j y_j = 0$  dan  $i = 1, 2, \dots, n$ .

Akan tetapi, apabila data yang digunakan tidak bisa dipisahkan dengan linier dapat menggunakan metode kernel. Dengan kernel, data *input space* dipetakan ke ruang berdimensi lebih tinggi. Menurut (Haykin, 1999) fungsi kernel yang umum dipakai ialah:

- 1) Kernel *linier*:  $x^T x$
- 2) Kernel *polynomial*:  $(x^T x + 1)^P$
- 3) Kernel *radial basis function* (RBF):  $\exp\left(-\frac{1}{2\sigma^2} \|x - x'\|^2\right)$
- 4) Kernel *sigmoid*:  $\tanh(\beta_0 x^T x_i + \beta_1)$

## 2.8 Confusion Matrix

*Confusion matrix* atau bisa juga disebut dengan *error matrix* merupakan metode yang umum digunakan untuk mengevaluasi kinerja dari model klasifikasi. *Confusion matrix* memberikan informasi mengenai frekuensi perilaku tertentu diprediksi dengan benar dan frekuensi perilaku tersebut diklasifikasi sebagai perilaku lain.

Pada penelitian “*Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle*” (Ruuska et al., 2018), dijelaskan bahwa metode validasi yang bagus untuk digunakan dalam klasifikasi diskrit adalah *confusion*

*matrix*. Selain itu, *confusion matrix* menyajikan informasi tambahan mengenai penyebab *errors* dan kepentingannya. Tabel *confusion matrix* dapat digambarkan seperti yang ditunjukkan pada Tabel 2.1.

Tabel 2.1 Confusion Matrix

		Nilai sebenarnya	
		True	False
Prediksi	True	TP (True Positive) <i>Correct result</i>	FP (False Positive) <i>Unexpected result</i>
	False	FN (False Negative) <i>Missing result</i>	TN (True Negative) <i>Correct absence of result</i>

*Confusion matrix* bisa dipakai guna melihat performa dari model yang sudah dibuat dengan *performance metrics*. Salah satu *performance metrics* yang umum dan sering digunakan adalah *accuracy metric*. Matriks *accuracy* dipakai guna lihat nilai akurasi dari penerapan model *machine learning* yang telah dibuat. *Accuracy* dapat didefinisikan sebagai berikut:

$$Accuracy = \frac{\text{Jumlah prediksi yang benar}}{\text{Jumlah keseluruhan prediksi}} \quad (2.8)$$

*Accuracy* merupakan rasio perbandingan informasi yang diprediksi benar oleh sistem dari keseluruhan informasi. *Accuracy* dapat diformulakan sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.9)$$

Keterangan:

- TP (*true positive*) : informasi bernilai positif yang benar diprediksi positif oleh sistem
- TN (*true negative*) : informasi bernilai negatif yang benar diprediksi negatif oleh sistem
- FP (*false positive*) : informasi bernilai negatif yang diprediksi positif oleh sistem
- FN (*false negative*) : informasi bernilai positif yang diprediksi negatif oleh sistem

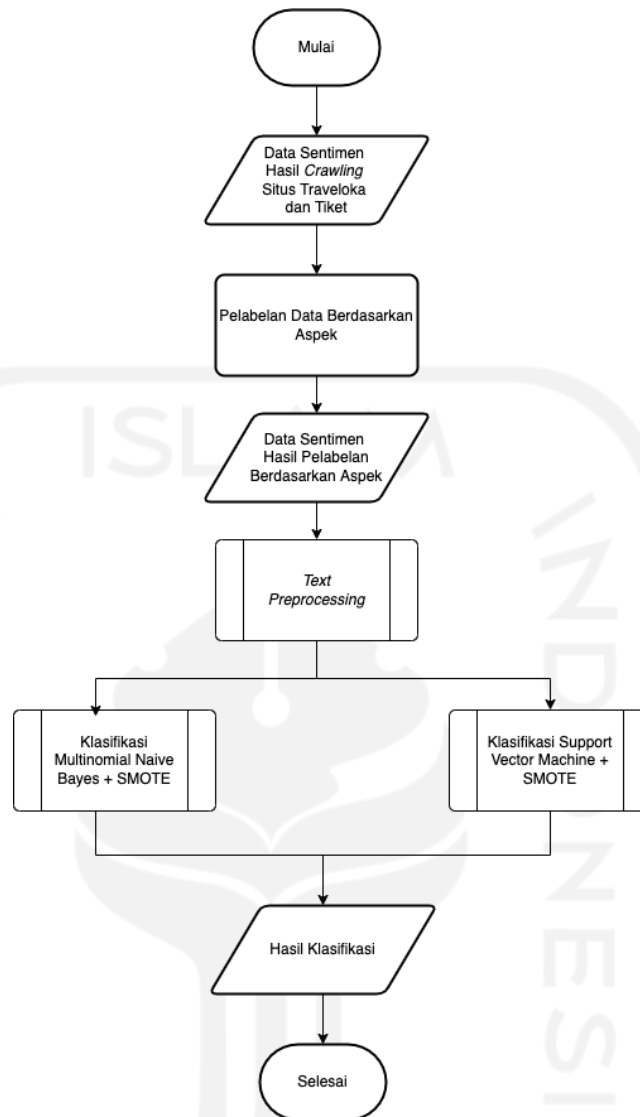
## **BAB III**

### **PELAKSANAAN MAGANG**

Selama menjalani kegiatan magang di Diskominfo DIY terdapat beberapa *task* yang dikerjakan. Tujuan *tasks* tersebut diberikan untuk membantu dalam pengembangan situs web Jogja Center. Berikut adalah *tasks* yang dikerjakan:

#### **3.1 Analisis Sentimen pada Ulasan Tempat Wisata Berdasarkan Aspek**

Pengunjung tempat wisata umumnya melihat ulasan dari beberapa situs yang menyediakan ulasan wisata terlebih dahulu sebelum mengunjungi tempat tersebut. Hal itu dilakukan untuk memastikan dahulu situs dan kondisi di tempat wisata yang ingin dikunjungi dari orang-orang yang sudah datang ke tempat itu. Ulasan-ulasan tempat wisata dari pengunjung dapat digunakan oleh pemerintah daerah setempat untuk melakukan evaluasi terkait tempat wisata yang berada di daerahnya. *Task* ini dikerjakan dengan tujuan untuk mengategorikan sentimen berdasarkan aspek sehingga dapat memudahkan Pemerintah DIY dalam mendapatkan umpan balik dari wisatawan mengenai tempat wisata yang dikelola. Bahasa pemrograman yang dipakai guna mengerjakan *task* ini ialah Python. Proses pengerjaan dalam *task* ini menggunakan Microsoft Excel untuk melakukan pelabelan dan Google Colab untuk pemrograman. Tahapan pengerjaan analisis sentimen pada tempat wisata berdasarkan aspek seperti yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 *Flowchart* Analisis Sentimen

### 3.1.1 Data Labelling

*Data labelling* atau pelabelan data merupakan tahapan awal yang dilakukan. *Labelling* merupakan proses memberi keterangan label sentimen (positif/netral/negatif) dari data ulasan tempat wisata. Data yang digunakan didapatkan dari hasil *crawling* situs web Traveloka dan Tiket. *Crawling* data dilakukan oleh *Data Engineer* Tim Jogja Center. Pada tahap ini, sentimen dikategorikan menjadi beberapa aspek yaitu *attraction*, *accessible*, *amenities*, *ancillary services* sebagaimana terlihat di Gambar 3.2. Jumlah data yang diberi label sebanyak 1922 data. Pengategorian aspek berdasarkan *guide* yang telah penulis buat sebelumnya. *Guide* sebagaimana terlihat di Gambar 3.3.

	A	B	C	D	E	F	G	H	I
1	id_review	nama_attrac	Judul_review	isi_review	attraction	acesable	amenities	ancillary services	
2	559761768	Museum Gunung Merapi	Jeepnya Mantapppppp..	MGM -Museum Gunung Merapi... di tempat ini, selain museum ada juga paket naik jeep alias lava tour. Naik jeep di MGM itu asyik banget... tempatnya parkirnya luas (buat bus muat).. ada toilet dan kantin. Sebelum jalan, ada briefing dari pihak MGM. Bisa request juga kalau kita takut atau ingin yang pelan-pelan aja jeepnya..atau rute yang pendek, semua stafnya ramah dan sopan, terima kasih banyak untuk mbak Ayi dan Mas Rryan atas semua bantuannya...	pos		0 pos	pos	
3	562830954	Tebing Breksi	Suitable for selfieholi	Arealnya sangat terbuka dan panas, disarankan bawa payung, cocok utk orang2 yg hobby selfie, fasilitas lengkap mulai dr area makanan dan ada masjid, tersedia jg ATV dan Jeep	neu		0 pos		0
4	580745671	Istana Air Tamansari	sebaiknya pakai tour guide agar tahu	ketika datang ke lokasi ini saya sarankan sewa lah tour gouide untuk membantu menjelaskan lokasi ini tanpa tour guide mungkin kita hanya melihat sebuah gedung tua yang kurang terawatw istana air tamansari sebenarnya adalah tempat pemandian dan aktivitas keluarga istana pada jaman dahulu, dibangun indah sebelum rumah penduduk banyak berdiri di sekeliling lokasi. apabila menggunakan tour guide, maka kita bisa menemukan cerita menarik mulai dari tempat mandi sultan, kisah perselingkuhan, tempat mereka berdinta, dapur, tempat sauna hingga design arsitektur awal yang sangat indah yakni sebuah bangunan di kelilingi danau buatan dengan akses melalui sebuah lorong di bawah tanah, sangat visioner pada jamannya	pos		0	0 pos	
5	570500370	Jalan Malioboro	Shopping	Mall di Malioboro Mall ini terletak di kawasan Malioboro depan depannya ada untuk tempat bersantai di dalamnya seperti biasa ya banyak barang branded juga tapi ga di sangka di sini masih ada store planet apa gitu lupa ( merk Quicksilver,ripicul,dll). Kalo malam depan mall ini suka ada yang ngamen musik lengkap gitu bagus. Kalo pagi di depan mall ini ada pecel yang enak dan rame banget. Awas jangan salah beli pecelnya pokoknya dia jam 6an udh buka kalo weekend jam 8 itu udah abis. Pokoknya bukan di depan McD ya tp paling ujung yg mau agak ngelewatn mall nya. Oke guys untuk para traveler jangan lupa kunjungi Jogja + makan pecelnya ya pasti nagih		0	0 pos		0

Gambar 3.2 Tabel *Labelling* Tempat Wisata

attraction	acesible	amenities	ancillary services
pengunjung tempat wisata menunjukkan poin mengapa tempat wisata menarik untuk dikunjungi	kemudahan dalam mencapai tempat wisata	akomodasi (hotel, motel, apartemen, dan lain-lain) terdekat	tourism information service
site attractions (tempat bersejarah, tempat dengan cuaca yang baik, tempat dengan pemandangan indah)	kondisi jalan menuju tempat wisata	penyediaan makanan dan minuman	profil wisata
event attractions (kejadian/peristiwa seperti kongres, pameran, dan lain-lain)	sarana transportasi yang tersedia	tempat beribadah	pemandu wisata
	jarak tempat wisata	sanitasi	
		aksesibilitas di tempat wisata (jalan setapak, pintu masuk, tempat parkir)	
		lainnya (dekat dengan gedung kantor/administrasi, pos keamanan, dan lain-lain)	

Gambar 3.3 *Guide* Pelabelan Tempat Wisata

Hasil dari distribusi data setelah dilakukan pelabelan sebagaimana ditunjukkan pada Tabel 3.1. Berdasarkan hasil tersebut, *dataset* yang telah diberi label memiliki banyak data yang bukan termasuk kategori pada aspek. Hal ini dikarenakan isi dari ulasan-ulasan tersebut tidak termasuk dalam aspek-aspek yang telah ditentukan.

Tabel 3.1 Hasil Data *Labelling*

Aspek	<i>Positive</i>	<i>Neutral</i>	<i>Negative</i>	Bukan kategori
sentiment_attraction	1313	75	24	510
sentiment_accessible	161	21	45	1695
sentiment_amenities	334	9	229	1350
sentiment_ancillary services	161	8	30	1723

### 3.1.2 Preprocessing

Tahap ini dilakukan untuk memudahkan model dalam mengolah data. Metode yang dilakukan disesuaikan berdasarkan *dataset* yang dimiliki. Metode-metode *preprocessing* yang dilakukan adalah:

#### 1. Case Folding

Tahap ini mengubah seluruh huruf ‘a’ - ‘z’ yang terdapat dalam dokumen jadi huruf kecil.

Contoh tahap *case folding* sebagaimana terlihat di Tabel 3.2.

Tabel 3.2 *Case Folding*

Sebelum	Sesudah
Pemandangan alamnya cukup bagus, bisa melihat sebagian Jogja dari tempat yang cukup tinggi	pemandangan alamnya cukup bagus, bisa melihat sebagian jogja dari tempat yang cukup tinggi
Mengunjungi tebing ini pada sore hari akan disugahi temaram senja yang teduh, hangat dan pemandangan	mengunjungi tebing ini pada sore hari akan disugahi temaram senja yang teduh, hangat dan pemandangan

#### 2. Filtering

Proses ini menghilangkan karakter selain huruf alfabet ‘a’-‘z’, yaitu seperti tanda baca, *hashtag*, lalu lain-lain. Contoh tahap *filtering* sebagaimana terlihat di Tabel 3.3.

Tabel 3.3 *Filtering*

Sebelum	Sesudah
pemandangan alamnya cukup bagus, bisa melihat sebagian jogja dari tempat yang cukup tinggi	pemandangan alamnya cukup bagus bisa melihat sebagian jogja dari tempat yang cukup tinggi
mengunjungi tebing ini pada sore hari akan disugahi temaram senja yang teduh, hangat dan pemandangan	mengunjungi tebing ini pada sore hari akan disugahi temaram senja yang teduh hangat dan pemandangan

### 3. Stop words removal

Proses menghilangkan kata-kata umum yang biasa ada pada jumlah yang cukup banyak dan dianggap tidak bermakna. Contoh tahap *stop words removal* sebagaimana terlihat di Tabel 3.4.

Tabel 3.4 *Stopwords Removal*

Sebelum	Sesudah
pemandangan alamnya cukup bagus bisa melihat sebagian jogja dari tempat yang cukup tinggi	pemandangan alamnya cukup bagus melihat sebagian jogja tempat cukup tinggi.
mengunjungi tebing ini pada sore hari akan disugahi temaram senja yang teduh hangat dan pemandangan	mengunjungi tebing sore hari disugahi temaram senja teduh hangat pemandangan

Sebelum dilakukan *pre-processing*, dilakukan *import library* dan *stop words* yang diperlukan terlebih dahulu sebagaimana terlihat di Gambar 3.4. *Library* yang digunakan di tahap ini adalah *regex* atau *regular expression*. Untuk daftar *stop words* yang digunakan tersusun dari 2 (dua) bahasa yakni Bahasa Inggris serta Bahasa Indonesia. Daftar *stop words* didapatkan dari mentor *Data Analyst* Tim Jogja Center. Adapun kode program yang digunakan untuk melakukan *pre-processing* ditunjukkan oleh Gambar 3.5.

```

import re

with open('/content/dict/id_stopwords.txt') as f:
    id_stopwords = f.read().splitlines()

with open('/content/dict/en_stopwords.txt') as f:
    en_stopwords = f.read().splitlines()

stopwords = id_stopwords + en_stopwords

```

Gambar 3.4 Kode Program Sebelum *Pre-processing*

```

1 def clean_text(text):
2     temp = str(text).lower()
3     temp = re.sub('\n', " ", temp)
4     temp = re.sub('\ ', " ", temp)
5     temp = re.sub('-', " ", temp)
6     temp = re.sub("[^a-z]", " ", str(temp))
7     temp = temp.split()
8     temp = [w for w in temp if not w in stopwords]
9     temp = " ".join(word for word in temp)
10
11     return temp
12
13 df1['cleaned_judul'] = df1['judul_review'].apply(clean_text)
14 df1['cleaned_isi'] = df1['isi_review'].apply(clean_text)
15 df1['cleaned'] = df1['cleaned_isi']+df1['cleaned_judul']

```

Gambar 3.5 Kode Program *Pre-processing*

Pada Gambar 3.5, kode pada baris 2 menunjukkan kode untuk *case folding*. Selanjutnya, baris 3-7 menunjukkan kode untuk *filtering*. Kemudian, baris 8 adalah kode menghilangkan *stopwords*.

### 3.1.3 Klasifikasi

#### a. Multinomial Naïve Bayes

Metode pertama yang digunakan dalam membuat model klasifikasi adalah algoritma Multinomial Naïve Bayes. Metode ini ialah suatu metode yang efektif digunakan dalam bidang klasifikasi teks. Di tahap ini, dilakukan pembagian *dataset* menjadi data *training* serta *testing* dengan rasio pembagian 80-20 (*training-testing*). Karena *dataset* yang dimiliki *imbalance*, dilakukan normalisasi dengan pendekatan *sampling* yaitu metode *oversampling*. Metode *oversampling* digunakan untuk menyeimbangkan jumlah distribusi dalam data dengan cara jumlah data di kelas minor ditingkatkan. Metode *oversampling* yang dilakukan menggunakan *Synthetic Minority Oversampling Technique* (SMOTE). SMOTE bekerja dengan membuat data *synthetic* baru dari kelas data minor. Dengan menggunakan SMOTE, distribusi data antara data kelas mayor dan minor menjadi seimbang seperti yang ditunjukkan pada Gambar 3.6. Setelah dilakukan normalisasi menggunakan SMOTE, tahap selanjutnya adalah membuat model klasifikasi. Berikut kode program yang digunakan untuk melakukan pemodelan sebagaimana terlihat di Gambar 3.7.



```

sentiment_attraction
before counter({'positive': 1052, 'na': 407, 'negative': 59, 'neutral': 19})
after counter({'positive': 1052, 'na': 1052, 'negative': 1052, 'neutral': 1052})

sentiment_accessible
before counter({'na': 1358, 'positive': 131, 'negative': 35, 'neutral': 13})
after counter({'na': 1358, 'positive': 1358, 'neutral': 1358, 'negative': 1358})

sentiment_amenities
before counter({'na': 1073, 'positive': 273, 'negative': 184, 'neutral': 7})
after counter({'na': 1073, 'positive': 1073, 'negative': 1073, 'neutral': 1073})

sentiment_ancillary_services
before counter({'na': 1385, 'positive': 127, 'negative': 21, 'neutral': 4})
after counter({'na': 1385, 'positive': 1385, 'neutral': 1385, 'negative': 1385})

```

Gambar 3.6 Distribusi Data Setelah Dilakukan SMOTE

```

1 cols = df1.iloc[:,4:8].columns
2 for col in aspects:
3     X = df1['cleaned']
4     y = y1[col]
5     print(col)
6
7     # split data
8     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 12)
9
10    # vectorizer
11    vectorizer = TfidfVectorizer(analyzer = 'char_wb', ngram_range=(2,3))
12    vectorizer.fit(X)
13    X_train_vect = vectorizer.fit_transform(X_train.apply(clean_text))
14    X_test_vect = vectorizer.transform(X_test.apply(clean_text))
15
16    # oversampling - smote
17    sm = SMOTE(k_neighbors=2, random_state=12)
18    X_train_vect, y_train = sm.fit_resample(X_train_vect, y_train)
19
20    # classification
21    mnb = MultinomialNB()
22    mnb.fit(X_train_vect, y_train)
23
24    # result
25    y_pred = mnb.predict(X_test_vect)
26    acc = accuracy_score(y_test, y_pred)
27    conf = confusion_matrix(y_test, y_pred)
28    print("Multinomial Naive Bayes "+str("{:04.2f}".format(acc*100))+'%')
29    print(conf)
30    print("\n")

```

Gambar 3.7 Kode Multinomial Naive Bayes

#### b. Support Vector Machine

Metode selanjutnya yang dipakai dalam membuat model klasifikasi adalah algoritma Support Vector Machine. Fungsi kernel yang akan dipakai adalah kernel linier. Kernel tersebut dipilih dikarenakan setelah mencoba menggunakan kernel *radial basis function* (RBF) dan *polynomial* menghasilkan *f1-score* yang belum cukup baik. Perbandingan hasil dari ketiga kernel tersebut seperti yang ditunjukkan pada Tabel 3.5, Tabel 3.6, dan Tabel 3.7.

Tabel 3.5 Hasil *F1-Score* SVM Kernel Linier

	<i>f1-score</i>			
	<b>attraction</b>	<b>accessible</b>	<b>amenities</b>	<b>ancillary services</b>
<b>na</b>	0,49	0,93	0,8	0,95
<b>negative</b>	0,17	0,31	0,56	0
<b>neutral</b>	0	0	0	0
<b>positive</b>	0,78	0,48	0,35	0,58

Tabel 3.6 Hasil *F1-Score* SVM Kernel RBF

	<i>f1-score</i>			
	<b>attraction</b>	<b>accessible</b>	<b>amenities</b>	<b>ancillary services</b>
<b>na</b>	0,35	0,94	0,86	0,95
<b>negative</b>	0	0,18	0,48	0
<b>neutral</b>	0	0	0	0
<b>positive</b>	0,82	0,18	0,35	0,37

Tabel 3.7 Hasil *F1-Score* SVM Kernel *Polynomial*

	<i>f1-score</i>			
	<b>attraction</b>	<b>accessible</b>	<b>amenities</b>	<b>ancillary services</b>
<b>na</b>	0,46	0,94	0,85	0,95
<b>negative</b>	0,11	0,18	0,52	0
<b>neutral</b>	0	0	0	0
<b>positive</b>	0,81	0,32	0,36	0,49

Pada tahap klasifikasi menggunakan metode SVM ini, dilakukan pembagian *dataset* jadi data *training* dan *testing* dengan rasio pembagian 80-20 (*training-testing*) dan normalisasi data *imbalance* menggunakan SMOTE. Berikut kode program yang digunakan untuk melakukan pemodelan sebagaimana terlihat di Gambar 3.8.

```

1 cols = df1.iloc[:,4:8].columns
2 for col in aspects:
3     X = df1['cleaned']
4     y = y1[col]
5     print(col)
6
7     # split data
8     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 12)
9
10    # vectorizer
11    vectorizer = TfidfVectorizer(analyzer = 'char_wb', ngram_range=(2,3))
12    vectorizer.fit(X)
13    X_train_vect = vectorizer.fit_transform(X_train.apply(clean_text))
14    X_test_vect = vectorizer.transform(X_test.apply(clean_text))
15
16    # oversampling - smote
17    sm = SMOTE(k_neighbors=3, random_state=42)
18    X_train_vect, y_train = sm.fit_resample(X_train_vect, y_train)
19
20    # svm
21    SVM = svm.SVC(C=1.5, kernel='linear', gamma=1.5)
22    SVM.fit(X_train_vect, y_train)
23    acc_score_svm = accuracy_score(SVM.predict(X_test_vect), y_test)
24    conf_svm = confusion_matrix(y_test, SVM.predict(X_test_vect))
25    print("Support Vector Machine "+str("{:04.2f}".format(acc_score_svm*100))+'%')
26    print(conf_svm)
27    print("\n")

```

Gambar 3.8 Kode Support Vector Machine

### 3.1.4 Evaluation

Tahap evaluasi model dilaksanakan untuk mengetahui performa dari model yang sudah dibuat sudah baik atau belum. Evaluasi model dapat dilakukan dengan melihat akurasi model melalui *confusion matrix*, hasil *accuracy* dan *f1-score*. Penggunaan *confusion matrix* memudahkan dalam perhitungan akurasi.

#### a. Multinomial Naïve Bayes

Hasil akurasi dan *f1-score* model menggunakan Multinomial Naïve Bayes sebagaimana terlihat di Tabel 3.8 dan Tabel 3.9.

Tabel 3.8 Hasil Akurasi Model Multinomial Naïve Bayes

Aspek	Akurasi
sentiment_attraction	62.60%
sentiment_accessible	61.82%
sentiment_amenities	55.58%
sentiment_ancillary_services	73.51%

Tabel 3.9 Hasil *F1-Score* Multinomial Naive Bayes

	<i>f1-score</i>			
	<b>attraction</b>	<b>acesible</b>	<b>amenities</b>	<b>ancillary services</b>
<b>na</b>	0,49	0,76	0,65	0,84
<b>negative</b>	0,18	0,2	0,42	0,12
<b>neutral</b>	0,29	0	0,67	0
<b>positive</b>	0,74	0,28	0,37	0,38

Berdasarkan hasil akurasi dan *f1-score* tersebut, model Multinomial Naïve Bayes menghasilkan nilai yang belum cukup baik. Dilihat dari hasil evaluasi *f1-score* dan hasil pelabelan data pada Tabel 3.1, model menghasilkan nilai *f1-score* yang belum cukup baik dikarenakan data yang dimiliki sebelum dilakukan normalisasi hanya sedikit. Untuk melihat ketepatan model dalam melakukan prediksi dapat menggunakan *confusion matrix*. Pada Tabel 3.10 ditunjukkan *confusion matrix* aspek *amenities*, dapat dilihat di tabel tersebut bahwa model Multinomial Naïve Bayes melakukan beberapa salah prediksi seperti 7 data dengan polaritas positif diprediksi sebagai data dengan polaritas negatif.

Tabel 3.10 *Confusion Matrix* Aspek *Amenities* Multinomial Naive Bayes

		<i>prediction values</i>			
		<b>na</b>	<b>negative</b>	<b>neutral</b>	<b>positive</b>
<i>actual values</i>	<b>na</b>	146	61	0	70
	<b>negative</b>	5	32	0	8
	<b>neutral</b>	0	1	1	0
	<b>positive</b>	19	7	0	35

#### b. Support Vector Machine

Hasil akurasi dan *f1-score* model menggunakan Support Vector Machine sebagaimana terlihat di Tabel 3.11 dan Tabel 3.12.

Tabel 3.11 Hasil Akurasi Model Support Vector Machine

<b>Aspek</b>	<b>Akurasi</b>
sentiment_attraction	66.49%
sentiment_acesible	87.27%
sentiment_amenities	69.35%
sentiment_ancillary_services	90.39%

Tabel 3.12 Hasil *F1-Score* Model Support Vector Machine

	<i>f1-score</i>			
	<b>attraction</b>	<b>accessible</b>	<b>amenities</b>	<b>ancillary services</b>
<b>na</b>	0,49	0,93	0,8	0,95
<b>negative</b>	0,17	0,31	0,56	0
<b>neutral</b>	0	0	0	0
<b>positive</b>	0,78	0,48	0,35	0,58

Berdasarkan hasil akurasi di atas, model Support Vector Machine menghasilkan nilai akurasi yang lebih baik daripada model Multinomial Naïve Bayes. Akan tetapi, untuk *f1-score* menghasilkan nilai yang belum cukup baik untuk sentimen *neutral*. Apabila dilihat dari hasil *f1-score* dan hasil pelabelan data pada Tabel 3.1, dapat ditarik kesimpulan yang sama seperti penggunaan dengan algoritma Multinomial Naïve Bayes bahwa karena data yang dimiliki sebelum dilakukan normalisasi sedikit sehingga nilai *f1-score* yang dihasilkan belum cukup baik. Selanjutnya dilihat dari hasil akurasi pada Tabel 3.11, aspek *attraction* dan *amenities* menghasilkan akurasi yang belum cukup baik. Hal tersebut ditunjukkan dari *confusion matrix* aspek *attraction*. Pada Tabel 3.13 ditunjukkan bahwa model SVM masih melakukan beberapa salah prediksi seperti 50 data bukan kategori diprediksi sebagai data dengan polaritas positif.

Tabel 3.13 *Confusion Matrix* Aspek *Attraction* Support Vector Machine

		<b>prediction values</b>			
		<b>na</b>	<b>negative</b>	<b>neutral</b>	<b>positive</b>
<b>actual values</b>	<b>na</b>	51	2	0	50
	<b>negative</b>	6	2	0	8
	<b>neutral</b>	1	1	0	3
	<b>positive</b>	56	2	0	203

### 3.1.5 Testing

Tahap *testing* model ini dilakukan untuk melihat apakah model sudah dapat memprediksi dengan tepat atau belum. Metode yang digunakan dalam melakukan *testing* ini adalah Support Vector Machine. Kode program yang digunakan dan hasil prediksi model sebagaimana ditunjukkan pada Gambar 3.9.

```

▶ for col in aspects:
  print(col)
  message_new = ['Cocok apabila liburan bersama anak kita ajak ke keraton Jogja. \
                Ada nilai wisata sejarah berdirinya Keraton Jogja serta nama nama sultan yang pernah bertahta. \
                Tiket murah pengetahuan bertambah. selamat berkunjung']
  X_new = vectorizer.transform(message_new)
  print(SVM.predict(X_new))

sentiment_attraction
['positive']
sentiment_accessible
['positive']
sentiment_amenities
['positive']
sentiment_ancillary_services
['positive']

```

Gambar 3.9 Kode Program dan Hasil Prediksi Model SVM

Berdasarkan Gambar 3.9, model Support Vector Machine dapat melakukan prediksi dengan tepat untuk aspek *attraction* dan *ancillary services*. Akan tetapi, model SVM salah melakukan prediksi untuk aspek *accessible* dan *amenities*.

### 3.2 Analisis Sentimen Media Sosial Twitter

Pada web Jogja Center terdapat dasbor “Sentimen Tempat Wisata Berdasarkan Data Media Sosial”. Pada dasbor tersebut, terdapat fitur komentar wisata berdasarkan media sosial. *Task* ini dikerjakan untuk mengecek sentimen yang diklasifikasikan negatif oleh model *machine learning* dengan kata kunci “Yogyakarta” yang diambil pada media sosial Twitter sudah akurat atau belum. Proses pengerjaan dalam *task* ini menggunakan Microsoft Excel dan Google Colab.

#### 3.2.1 Data Labelling

Dilakukan pengecekan data dengan melakukan *labelling* secara manual. Data yang digunakan adalah data media sosial Twitter yang memiliki kata kunci “Yogyakarta” berjumlah 4684 data. *Labelling* dilakukan berdasarkan *guideline* yang telah diberikan. *Guideline* pelabelan terlihat di Gambar 3.10. Data dan *guideline* didapatkan dari mentor *Data Analyst* Tim Jogja Center. Proses *labelling* terlihat di Gambar 3.11.

Labelling Rules and Guidelines Sentiment Tweet

File Edit View Tools Help

Request edit access

Share

**Labelling Rules and Guidelines**

1. Sentiment: **positive, neutral, negative, skip**

**Sentiment Rules**

1. Dilabeli positif ketika menunjukkan emosi yang positif
2. Dilabeli netral ketika tidak ada emosi yang ditunjukkan, misalnya hanya berupa informasi, anjuran, ajakan
3. Dilabeli negatif apabila ada kata umpatan/makian, berupa ancaman, kesan dan yang emosi yang negatif, dan semacamnya
4. Jika isi post adalah hal yang dirasa kurang relevan, misal kpop atau percakapan sejenis, maka skip
5. Labeli dengan konsisten
6. Jika ragu dengan label yang tepat, maka skip
7. Jika bahasanya campur dengan bahasa lain (bahasa daerah, bahasa asing), dan bahasa asing/daerahnya lebih dominan, maka skip
8. Jika post hanya berisi hashtag, links, mentions, atau emoticons, maka diskip saja

Gambar 3.10 *Labelling Rules dan Guidelines*

	A	B	C	D	E	F	G
1	link	user.screen_message	Sri Sultan HB X Minta Aparat Tindak Tegas Pelanggar Prokes - Jawa Tengah - <a href="https://t.co/2KmDvcDWUZ">https://t.co/2KmDvcDWUZ</a> <a href="https://t.co/TQpbGRoQ2d">https://t.co/TQpbGRoQ2d</a>	sentiment	note		
2	<a href="https://twit/JogjaTrans2">https://twit/JogjaTrans2</a>	#jogja #sewamobil #rentalmobil #sewamobiljogja #rentalmobiljogja #wisatajogja #mobilwisatajogja #diy #yogya #yogyakarta #jogjakarta	Kronologi Pria Misterius Serang Mapolresta Yogya,Acungkan Parang ke Polisi hingga Akhirnya Diringkus - Tribun Jogja <a href="https://t.co/ODMata9snz">https://t.co/ODMata9snz</a>	neutral	berita		
3	<a href="https://twit/JogjaTrans2">https://twit/JogjaTrans2</a>	#jogja #sewamobil #rentalmobil #sewamobiljogja #rentalmobiljogja #wisatajogja #mobilwisatajogja #diy #yogya #yogyakarta #jogjakarta	Pelanggaran Prokes Masih Terjadi di Area Wisata - Harian Jogja <a href="https://t.co/1Dxd1ETxry">https://t.co/1Dxd1ETxry</a>	negative	berita		
4	<a href="https://twit/JogjaTrans2">https://twit/JogjaTrans2</a>	#jogja #sewamobil #rentalmobil #sewamobiljogja #rentalmobiljogja #wisatajogja #mobilwisatajogja #diy #yogya #yogyakarta #jogjakarta	Corona Ngegas di Yogya: Ruang Isolasi-RS Penuh, Stok Oksigen Menipis - detikNews <a href="https://t.co/0d0EKhsdjE">https://t.co/0d0EKhsdjE</a>	negative	berita		

Gambar 3.11 Tabel *Labelling* Sentimen Negatif

Selanjutnya, dilihat persebaran data dari *dataset* yang telah diberi label menggunakan Python. Hasil persebaran data terlihat di Gambar 3.12.

```
[6] temp = df['sentiment'].value_counts()
print(temp)

neutral    2396
negative   1935
positive    283
-           70
Name: sentiment, dtype: int64
```

Gambar 3.12 Persebaran Data Media Sosial

### 3.2.2 Preprocessing

Tahap *pre-processing* dilakukan menggunakan bahasa pemrograman Python. Metode-metode *pre-processing* yang akan dilakukan adalah *case folding*, *filtering*, dan *stop words removal*. Tahap ini akan membuat *dataset* yang digunakan menjadi lebih rapi dan mudah untuk diolah pada proses pemodelan. Berikut kode program yang digunakan dalam melakukan *pre-processing* sebagaimana terlihat Gambar 3.4 dan Gambar 3.5.

### 3.2.3 Klasifikasi

Metode yang digunakan dalam membuat model klasifikasi ini terdapat beberapa algoritma yang dicoba yaitu Linear SVC, Maximum Entropy, Support Vector Machine, serta Multinomial Naïve Bayes. Dilaksanakan klasifikasi memakai beberapa algoritma untuk melihat algoritma mana yang menghasilkan akurasi yang cukup baik. Adapun kode program yang digunakan guna lakukan pemodelan sebagaimana terlihat di Gambar 3.13. Berdasarkan Gambar 3.10, baris 15-21 menunjukkan kode program untuk metode Linear SVC, baris 24-35 untuk metode Maximum Entropy, baris 33-39 untuk metode Support Vector Machine, dan baris 42-47 untuk metode Multinomial Naïve Bayes.



```

1 cols = df.iloc[:,3:4].columns
2
3 X = df['cleaned_message']
4 y = df['sentiment']
5
6 # split
7 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
8         random_state=100)
9
10 # vectorizer
11 vectorizer = TfidfVectorizer()
12 X_train_vect = vectorizer.fit_transform(X_train.apply(clean_text))
13 X_test_vect = vectorizer.transform(X_test.apply(clean_text))
14
15 # svc
16 SVC = LinearSVC()
17 SVC.fit(X_train_vect, y_train)
18 acc_score_svc = accuracy_score(SVC.predict(X_test_vect), y_test)
19 conf_svc = confusion_matrix(y_test, SVC.predict(X_test_vect))
20 print("Linear Support Vector Classification: "+str("
21     {:04.2f}".format(acc_score_svc*100))+'%')
22 print(conf_svc)
23 print("\n")
24
25 # maxent
26 maxent = LogisticRegression(C=1, multi_class='multinomial', solver='saga',
27     penalty = 'l2',max_iter = 50000)
28 maxent.fit(X_train_vect, y_train)
29 acc_score_maxent = accuracy_score(maxent.predict(X_test_vect), y_test)
30 conf_maxent = confusion_matrix(y_test, maxent.predict(X_test_vect))
31 print("Maximum Entropy: "+str("{:04.2f}".format(acc_score_maxent*100))+'%')
32 print(conf_maxent)
33 print("\n")
34
35 # svm
36 SVM = svm.SVC(C=1.5, kernel='rbf', gamma=1.5)
37 SVM.fit(X_train_vect, y_train)
38 acc_score_svm = accuracy_score(SVM.predict(X_test_vect), y_test)
39 conf_svm = confusion_matrix(y_test, SVM.predict(X_test_vect))
40 print("Support Vector Machine: "+str("
41     {:04.2f}".format(acc_score_svm*100))+'%')
42 print(conf_svm)
43 print("\n")
44
45 # mnb
46 MNB = MultinomialNB()
47 MNB.fit(X_train_vect, y_train)
48 acc_score_mnb = accuracy_score(MNB.predict(X_test_vect), y_test)
49 conf_mnb = confusion_matrix(y_test, MNB.predict(X_test_vect))
50 print("Multinomial Naive Bayes: "+str("
51     {:04.2f}".format(acc_score_mnb*100))+'%')
52 print(conf_mnb)

```

Gambar 3.13 Kode Program *Modelling*

### 3.2.4 Evaluation

Tahap ini dipakai guna melihat akurasi dari model yang dibuat sudah baik atau belum. Evaluasi model dilakukan melalui hasil akurasi dan *confusion matrix* terlihat di Gambar 3.14.

```

Linear Support Vector Classification: 76.92%
[[261  65   2]
 [119 420   7]
 [  4  16 29]]

Maximum Entropy: 79.52%
[[263  65   0]
 [ 95 450   1]
 [  4  24 21]]

Support Vector Machine: 80.61%
[[264  64   0]
 [ 89 456   1]
 [  4  21 24]]

Multinomial Naive Bayes: 76.38%
[[238  90   0]
 [ 88 458   0]
 [  4  36   9]]

```

Gambar 3.14 Hasil Evaluasi Sentimen Negatif

Berdasarkan akurasi dan *confusion matrix* pada Gambar 3.11, algoritma yang menghasilkan akurasi paling baik adalah Support Vector Machine dengan akurasi 80.61%. Algoritma SVM dapat melakukan prediksi dengan benar 264 data negatif, 456 data netral, dan 24 data positif

### 3.3 Task Kolaborasi

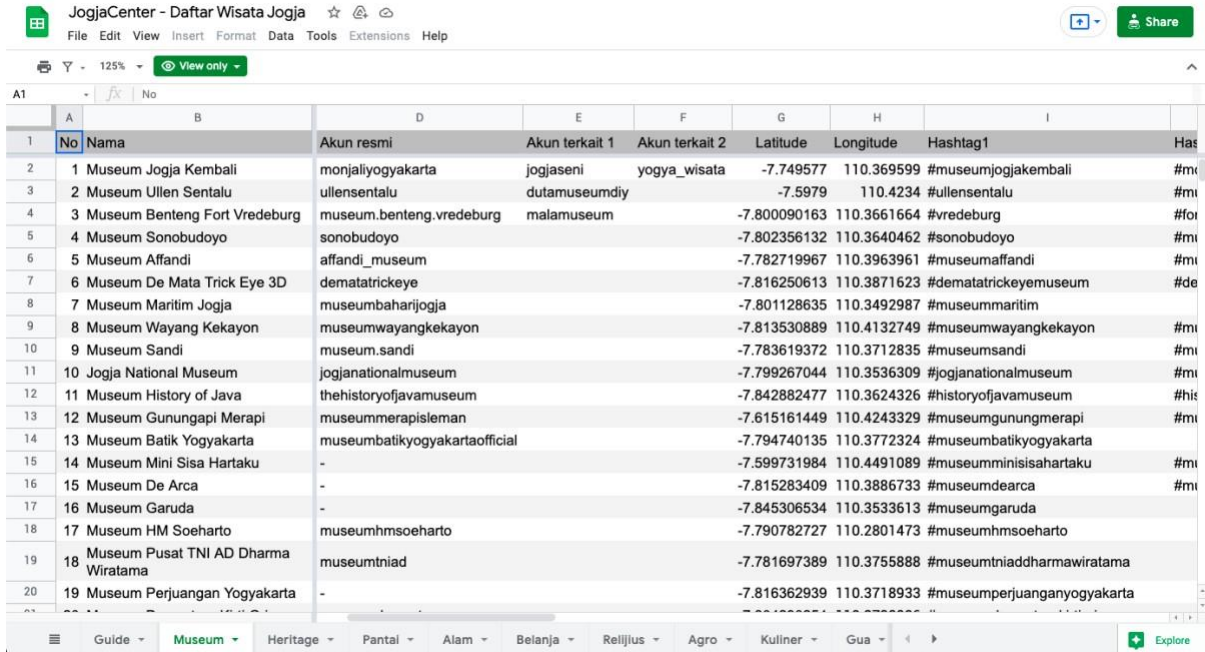
Selama pelaksanaan magang di Diskominfo DIY, terdapat beberapa *task* diberikan yang dikerjakan bersama peserta magang lainnya, yaitu:

#### 3.3.1 List Data Tempat Wisata di Yogyakarta

Yogyakarta memiliki banyak tempat wisata yang menarik untuk dikunjungi. Namun, dari bermacam tempat wisata tersebut banyak juga yang masih belum dikenal oleh pengunjung tempat wisata. Tujuan dari *task* ini diberikan untuk keperluan data di Jogja Center, mendata tempat-tempat wisata yang terdapat di Yogyakarta.

*Task* ini dikerjakan dengan membuat 10 (sepuluh) jenis kategori tempat wisata (museum, *heritage*, pantai, alam, belanja, religius, kuliner, agro, gua, dan budaya). Kategori ini dibuat untuk memudahkan saat pendataan. Beban pekerjaan dibagi dengan adil bersama peserta magang lain. Penulis mendapat kategori wisata museum, *heritage*, dan budaya.

Proses pengerjaan *task* ini dengan melakukan pencarian di internet dan media sosial menggunakan kata kunci yang dibutuhkan. Data-data yang dicari adalah nama tempat wisata, akun media sosial resmi, akun-akun terkait, *latitude*, *longitude*, dan *hashtag* yang berkaitan dengan tempat wisata tersebut. Hasil pengerjaan *task* ini berupa daftar tempat-tempat wisata yang berada di Yogyakarta sebagaimana ditunjukkan pada Gambar 3.15. *Task* ini dikerjakan menggunakan Google Sheet untuk memudahkan dalam pengerjaan.



No	Nama	Akun resmi	Akun terkait 1	Akun terkait 2	Latitude	Longitude	Hashtag1	Has
1	Museum Jogja Kembali	monjallyogyakarta	jogjaseni	yogya_wisata	-7.749577	110.369599	#museumjogjakembali	#mi
2	Museum Ullen Sentalu	ullensentalu	dutamuseumdiy		-7.5979	110.4234	#ullensentalu	#mi
3	Museum Benteng Fort Vredeburg	museum.benteng.vredeburg	malamuseum		-7.800090163	110.3661664	#vredeburg	#fo
4	Museum Sonobudoyo	sonobudoyo			-7.802356132	110.3640462	#sonobudoyo	#mi
5	Museum Affandi	affandi_museum			-7.782719967	110.3963961	#museumaffandi	#mi
6	Museum De Mata Trick Eye 3D	dematatickeye			-7.816250613	110.3871623	#dematatickeyemuseum	#de
7	Museum Maritim Jogja	museumbaharjogja			-7.801128635	110.3492987	#museummaritim	
8	Museum Wayang Kekayon	museumwayangkekayon			-7.813530889	110.4132749	#museumwayangkekayon	#mi
9	Museum Sandi	museum.sandi			-7.783619372	110.3712835	#museumsandi	#mi
10	Jogja National Museum	jogjanationalmuseum			-7.799267044	110.3536309	#jogjanationalmuseum	#mi
11	Museum History of Java	thehistoryofjavamuseum			-7.842882477	110.3624326	#historyofjavamuseum	#his
12	Museum Gunungapi Merapi	museummerapisleman			-7.615161449	110.4243329	#museumgunungmerapi	#mi
13	Museum Batik Yogyakarta	museumbatikyogyakartaofficial			-7.794740135	110.3772324	#museumbatikyogyakarta	
14	Museum Mini Sisa Hartaku	-			-7.599731984	110.4491089	#museumminisihartaku	#mi
15	Museum De Arca	-			-7.815283409	110.3886733	#museumdearca	#mi
16	Museum Garuda	-			-7.845306534	110.3533613	#museumgaruda	
17	Museum HM Soeharto	museumhmsoeharto			-7.790782727	110.2801473	#museumhmsoeharto	
18	Museum Pusat TNI AD Dharma Wiratama	museumtniad			-7.781697389	110.3755888	#museumtniaddharmawiratama	
19	Museum Perjuangan Yogyakarta	-			-7.816362939	110.3718933	#museumperjuanganyogyakarta	

Gambar 3.15 List Tempat Wisata

### 3.3.2 List Data Putusan Pengadilan di Yogyakarta

Tujuan diberikannya *task* ini adalah untuk keperluan data Jogja Center bagian sosial. Data yang digunakan adalah direktori putusan pengadilan negeri seta pengadilan agama di Yogyakarta. *Task* ini dikerjakan dengan membagi beban pekerjaan secara adil.

Pengerjaan *task* ini dengan mengambil beberapa data yang dibutuhkan yaitu: tanggal perkara, jenis perkara, nomor pasal KUHAP yang terkait, dan duduk perkara putusan. Proses pengerjaan *task* ini dengan cara manual dan *coding* menggunakan bahasa pemrograman Python. Pengerjaan dengan Python menggunakan *library* PyMuPDF, fitz, glob, dan nltk. Adapun kode program terlihat di Gambar 3.16.

```

# import library
import fitz
import glob
from nltk.tokenize import sent_tokenize

# mengambil data yang dibutuhkan
from google.colab import drive
drive.mount("/content/gdrive")

file_dir = "data/Direktori Putusan/Sleman/PN/"
files = glob.glob(file_dir+"*.pdf")

# tanggal, jenis perkara, nomer pasal KUHAP yg terkait, duduk perkara
# jenis perkara: nama file (pid: pidana, pdt: perdata)
# pasal KUHAP: diambil dari kalimat yg mengandung kata KUHAP
# duduk perkara: diambil dari kalimat yang mengandung kata "melakukan tindak
pidana"

for fl in files:
    if 'pdt' in fl.lower():
        jenis_perkara = "perdata"
    elif 'pid' in fl.lower():
        jenis_perkara = "pidana"

    doc = fitz.open(fl)
    all_text = ""
    for page in doc:
        text = page.get_text()
        all_text += text

    sentences = sent_tokenize(all_text)

    kuhap = []
    perkara = []
    for sen in sentences:
        if "kuhap" in sen.lower():
            kuhap.append(sen)
        if "melakukan tindak pidana" in sen.lower():
            perkara.append(sen)
    print(fl)
    print(jenis_perkara)
    print(kuhap)
    print(perkara)

```

Gambar 3.16 Kode Program *Task* Putusan Pengadilan

Hasil dari pengerjaan *task* ini berupa daftar putusan pengadilan di Yogyakarta sebagaimana yang ditunjukkan pada Gambar 3.17. *List* data ini dapat digunakan sebagai acuan dalam membuat *insight* Analisis Perkara Menurut Data Pengadilan pada dasbor Jogja Center bagian sosial. *Task* ini dikerjakan menggunakan Google Sheet.

file_name	tanggal	jenis_perkara	pasal	duduk_perkara
1-Pdt.P-2021-PN_Smn.pdf	18 Januari 2021	Perdata	UU No 23 tahun 2006 jo PP No 25 tahun 2008	Perbaikan Kesalahan Dalam Akta Kelahiran
1-Pid.C-2021-PN_Smn.pdf	5 Februari 2021	Pidana	Pasal 364 KUH Pidana Jo Pasal 53 KUH Pidana dan Peraturan Mahkamah Agung Republik Indonesia Nomor 02 tahun 2012 serta UU No. 8 tahun 1981	Pencurian
1-Pid.S-2021-PN_Smn.pdf	13 Januari 2021	Pidana	Pasal 37 Perda Kabupaten Sleman Nomor 08 Tahun 2019, Pasal 55 ayat ke-1 KUHP	Melakukan Peredaran dan Penjualan Minuman Beraikohol Tanpa Izin Secara Bersama-sama
10-Pdt.G-2021-PN_Smn.pdf	23 Februari 2021	Perdata	Pasal 271 jo 272 RV	Perbuatan Melawan Hukum
10-Pdt.P-2021-PN_Smn.pdf	28 Januari 2021	Perdata	UU No 24 tahun 2013	Permohonan Ganti Nama
11-Pdt.P-2021-PN_Smn.pdf	28 Januari 2021	Perdata		Permohonan Ganti Nama
12-Pdt.P-2021-PN_Smn.pdf	21 Januari 2021	Perdata		Permohonan Ganti Nama
121-Pdt.G-2020-PN_Smn.pdf	7 Januari 2021	Perdata	Pasal 271 jo 272 RV	Perbuatan Melawan Hukum
125-Pdt.G-2020-PN_Smn.pdf	4 Januari 2021	Perdata		Perbuatan Melawan Hukum
13-Pdt.P-2021-PN_Smn.pdf	3 Februari 2021	Perdata	UU No 23 tahun 2006	Permohonan Ganti Nama
13-Pid.B-2021-PN_Smn.pdf	1 Maret 2021	Pidana	Pasal 362 KUHP	Pencurian
14-Pdt.G.S-2020-PN_Smn.pdf	21 Januari 2021	Perdata		Perbuatan Melawan Hukum
14-Pdt.P-2021-PN_Smn.pdf	2 Februari 2021	Perdata	Pasal 52 UU No 23 tahun 2006 Jo UU No 24 tahun 2013 Pedoman Pelaksanaan Tugas dan	Permohonan Ganti Nama

Gambar 3.17 List Putusan Pengadilan

### 3.3.3 Report Korelasi Pergerakan Google Mobility dan Kenaikan Kasus COVID-19 di Yogyakarta

*Task* ini diberikan bertujuan untuk melihat korelasi pergerakan antara pergerakan masyarakat berdasarkan Google Mobility dengan kenaikan kasus COVID-19 di Yogyakarta. *Report* ini digunakan untuk acuan dalam membuat *insight* data vaksin COVID-19 pada dasbor Jogja Center bagian kesehatan. Pengerjaan *task* ini menggunakan Datawrapper untuk membuat visualisasinya. Tahapan-tahapan pengerjaan *task report* korelasi pergerakan ini yakni:

#### 1. Pengambilan data

- Data Google Mobility

Data Google Mobility dapat diperoleh dari web <https://www.google.com/covid19/mobility/>. Data tersebut diperbaharui setiap hari oleh Google. Data yang digunakan hanya data *regional* saja sehingga pengunduhan dilakukan dengan memilih *Region* CSVs. Data yang diunduh memiliki format *file* zip, ambil *file* csv Indonesia dengan nama ID. *File* csv berisi data dari tanggal 1 Januari 2021, untuk *task* ini penulis hanya mengambil data dari bulan Mei-Juli 2021. Data seperti yang ditunjukkan pada Gambar 3.18.

Name	Date Modified	Size	Kind
2021_GW_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	20 KB	Comma...et (.csv)
2021_HK_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	28 KB	Comma...et (.csv)
2021_HN_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	2 MB	Comma...et (.csv)
2021_HR_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	765 KB	Comma...et (.csv)
2021_HT_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	304 KB	Comma...et (.csv)
2021_HU_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	690 KB	Comma...et (.csv)
2021_ID_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	1,2 MB	Comma...et (.csv)
2021_IE_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	888 KB	Comma...et (.csv)
2021_IL_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	676 KB	Comma...et (.csv)
2021_IN_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	21,8 MB	Comma...et (.csv)
2021_IQ_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	26 KB	Comma...et (.csv)
2021_IT_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	4,8 MB	Comma...et (.csv)
2021_JM_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	489 KB	Comma...et (.csv)
2021_JO_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	458 KB	Comma...et (.csv)
2021_JP_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	1,4 MB	Comma...et (.csv)
2021_KE_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	1,2 MB	Comma...et (.csv)
2021_KG_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	268 KB	Comma...et (.csv)
2021_KH_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	763 KB	Comma...et (.csv)
2021_KR_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	64 KB	Comma...et (.csv)
2021_KW_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	290 KB	Comma...et (.csv)
2021_KZ_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	29 KB	Comma...et (.csv)
2021_LA_Region_Mobility_Report.csv	14 Jan 2022 1:38 PM	27 KB	Comma...et (.csv)

Gambar 3.18 Data Google Mobility

- Data kasus COVID-19 di Yogyakarta

Data kasus COVID-19 didapatkan dari @kawalcovid19. Data yang diperlukan untuk *task* ini hanya tren harian COVID-19 terkonfirmasi di Yogyakarta. Data yang dipakai ialah data bulan Mei-Juli 2021 seperti yang ditunjukkan pada Gambar 3.19.

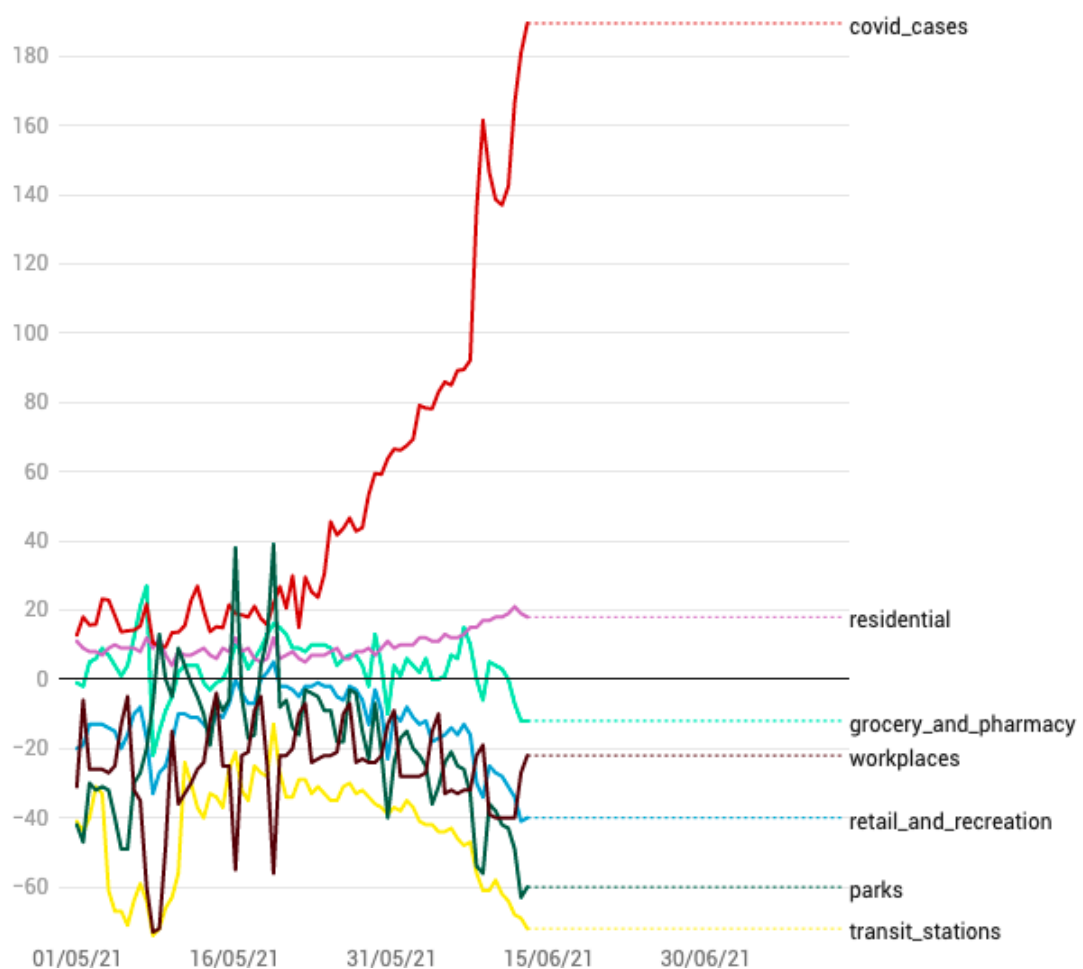
	A	E	F	G	H	I	J	K	L	M	N	O	S	T	U	X
1		Total kasus	Kasus aktif	% kasus aktif	Sembuh (baru)	Sembuh	Tingkat kesembuhan (seluruh)	Tingkat kesembuhan (closed)	Meninggal (baru)	Meninggal Dunia	Tingkat kematian (seluruh)	Tingkat kematian (closed)	Jumlah spesimen diperiksa	Jumlah orang diperiksa	Negatif	Spesime
61	1-May	10,551	8,160	77.34%	69	1,591	15.08%	66.54%	8	800	7.58%	33.46%	102,305	76,538	65,987	7,706
62	2-May	10,843	8,347	76.98%	74	1,665	15.36%	66.71%	31	831	7.66%	33.29%	107,943	79,868	69,025	5,638
63	3-May	11,192	8,471	75.69%	211	1,876	16.76%	68.95%	14	845	7.55%	31.05%	112,965	83,012	71,820	5,022
64	4-May	11,587	8,769	75.68%	78	1,954	16.86%	69.34%	19	864	7.46%	30.66%	116,861	86,061	74,474	3,896
65	5-May	12,071	9,002	74.58%	243	2,197	18.20%	71.59%	8	872	7.22%	28.41%	121,547	88,924	76,853	4,686
66	6-May	12,438	9,226	74.18%	120	2,317	18.63%	72.14%	23	895	7.20%	27.86%	128,383	92,976	80,538	6,836
67	7-May	12,776	9,465	74.08%	64	2,381	18.64%	71.91%	35	930	7.28%	28.09%	134,151	96,717	83,941	5,768
68	8-May	13,112	9,675	73.79%	113	2,494	19.02%	72.56%	13	943	7.19%	27.44%	143,452	103,361	90,249	9,301
69	9-May	13,645	10,079	73.87%	113	2,607	19.11%	73.11%	16	959	7.03%	26.89%	150,887	108,899	95,054	7,435
70	10-May	14,032	10,361	73.84%	91	2,698	19.23%	73.49%	14	973	6.93%	26.51%	158,273	113,452	99,420	7,386
71	11-May	14,285	10,393	72.86%	183	2,881	20.20%	74.41%	18	991	6.95%	25.59%	161,351	116,358	102,093	3,078
72	12-May	14,749	10,679	72.40%	182	3,063	20.77%	75.26%	16	1,007	6.83%	24.74%	165,128	119,728	104,579	3,777
73	13-May	15,438	11,123	72.05%	224	3,287	21.29%	76.18%	21	1,028	6.66%	23.82%	169,195	123,572	108,134	4,067
74	14-May	16,006	11,445	71.50%	231	3,518	21.98%	77.13%	15	1,043	6.52%	22.87%	173,690	127,813	111,807	4,495
75	15-May	16,496	11,617	70.42%	285	3,803	23.05%	77.95%	33	1,076	6.52%	22.05%	178,602	132,060	115,564	4,912
76	16-May	17,025	12,025	70.63%	108	3,911	22.97%	78.22%	13	1,089	6.40%	21.78%	182,818	135,725	118,700	4,216
77	17-May	17,514	12,237	69.87%	218	4,129	23.58%	78.25%	59	1,148	6.55%	21.75%	187,965	140,473	122,959	5,147
78	18-May	18,010	12,495	69.38%	195	4,324	24.01%	78.40%	43	1,191	6.61%	21.60%	190,660	143,035	125,025	2,895
79	19-May	18,496	12,808	69.25%	143	4,467	24.15%	78.53%	30	1,221	6.60%	21.47%	202,936	147,799	129,303	12,276
80	20-May	19,189	13,372	69.69%	108	4,575	23.84%	78.65%	21	1,242	6.47%	21.35%	211,883	154,139	134,950	8,947
81	21-May	20,162	14,046	69.67%	263	4,838	24.00%	79.10%	36	1,278	6.34%	20.90%	219,975	160,374	140,212	8,092
82	22-May	20,796	14,413	69.31%	219	5,057	24.32%	79.23%	48	1,326	6.38%	20.77%	229,334	168,969	148,173	9,359
83	23-May	21,745	15,145	69.65%	192	5,249	24.14%	79.53%	25	1,351	6.21%	20.47%	239,740	176,035	154,290	10,406
84	24-May	22,271	15,497	69.58%	153	5,402	24.26%	79.75%	21	1,372	6.16%	20.25%	248,555	179,864	157,593	8,815

Gambar 3.19 Data kasus COVID-19

## 2. Pengolahan data

Data yang sudah dipat diolah memakai Datawrapper untuk dijadikan visualisasi dalam bentuk grafik. Visualisasi ini dibuat untuk memudahkan dalam melihat korelasi pergerakannya. Untuk mempermudah dalam pembacaan grafik karena terdapat skala data yang berbeda, data COVID-19 dibagi menjadi skala 1:10. Hasil visualisasi grafik terlihat di Gambar 3.20. Berdasarkan hasil itu memperlihatkan ketika kasus COVID-19 sedang meningkat, masyarakat lebih banyak yang berada di rumah saja ditunjukkan dengan grafik *residential* yang naik.

### Comparing between Mobility and Covid Cases in Yogyakarta



Gambar 3.20 Grafik Komparasi

### 3.3.4 Report Vaksinasi di Yogyakarta

Tujuan *task* ini diberikan untuk melihat akumulasi masyarakat yang sudah melakukan vaksinasi baik itu dosis pertama dan kedua di Yogyakarta. Untuk memudahkan dalam melihat

akumulasi dibuat visualisasi dalam bentuk grafik. Pengerjaan *task* ini menggunakan Datawrapper. Tahapan-tahapan pengerjaan *task report* korelasi pergerakan ini yakni:

### 1. Pengambilan data

Data vaksinasi didapatkan lewat <https://vaksin.kemkes.go.id/>. Data tersebut diperbaharui setiap harinya oleh Kemenkes RI. Data yang didapatkan berupa data harian jumlah masyarakat yang melakukan vaksinasi baik itu dosis pertama maupun kedua di tiap kabupaten serta kota di Yogyakarta. Untuk *task* ini, data yang dipakai di bulan Mei-Juli 2021 sebagaimana ditunjukkan di Gambar 3.21.

Provinsi	Tanggal	Bantul	Gunungkidul	Kulon Progo	Sleman	Yogyakarta
Daerah Istimewa Yogyakarta	01/05/21	111	0	42	56	0
Daerah Istimewa Yogyakarta	02/05/21	0	0	0	0	1
Daerah Istimewa Yogyakarta	03/05/21	1326	445	74	2490	816
Daerah Istimewa Yogyakarta	04/05/21	4221	385	582	2205	498
Daerah Istimewa Yogyakarta	05/05/21	2073	534	685	2486	1693
Daerah Istimewa Yogyakarta	06/05/21	1467	615	861	2802	2665
Daerah Istimewa Yogyakarta	07/05/21	763	203	428	1963	2425
Daerah Istimewa Yogyakarta	08/05/21	1001	61	385	772	453
Daerah Istimewa Yogyakarta	09/05/21	0	0	0	0	0
Daerah Istimewa Yogyakarta	10/05/21	1424	414	166	1662	534
Daerah Istimewa Yogyakarta	11/05/21	191	361	395	1157	551
Daerah Istimewa Yogyakarta	12/05/21	0	49	89	189	194
Daerah Istimewa Yogyakarta	13/05/21	0	0	0	1	0
Daerah Istimewa Yogyakarta	14/05/21	0	0	0	0	0
Daerah Istimewa Yogyakarta	15/05/21	91	217	0	0	241
Daerah Istimewa Yogyakarta	16/05/21	0	0	0	0	0
Daerah Istimewa Yogyakarta	17/05/21	1124	449	176	1085	1034
Daerah Istimewa Yogyakarta	18/05/21	1076	1563	865	2995	1060
Daerah Istimewa Yogyakarta	19/05/21	975	1137	804	3973	2613
Daerah Istimewa Yogyakarta	20/05/21	1094	1323	1057	4694	2422
Daerah Istimewa Yogyakarta	21/05/21	1161	373	495	1123	580
Daerah Istimewa Yogyakarta	22/05/21	326	634	504	2073	1845
Daerah Istimewa Yogyakarta	23/05/21	9	0	0	1	2
Daerah Istimewa Yogyakarta	24/05/21	1348	749	41	2609	675
Daerah Istimewa Yogyakarta	25/05/21	1200	906	727	2397	1047
Daerah Istimewa Yogyakarta	26/05/21	0	18	0	80	0
Daerah Istimewa Yogyakarta	27/05/21	3005	1165	1036	3415	1466
Daerah Istimewa Yogyakarta	28/05/21	2913	360	561	346	599
Daerah Istimewa Yogyakarta	29/05/21	1510	35	569	13	377
Daerah Istimewa Yogyakarta	30/05/21	5	0	0	8	388
Daerah Istimewa Yogyakarta	31/05/21	718	529	202	631	472

Gambar 3.21 Data Vaksinasi Dosis Kedua



## 2. Pengolahan data

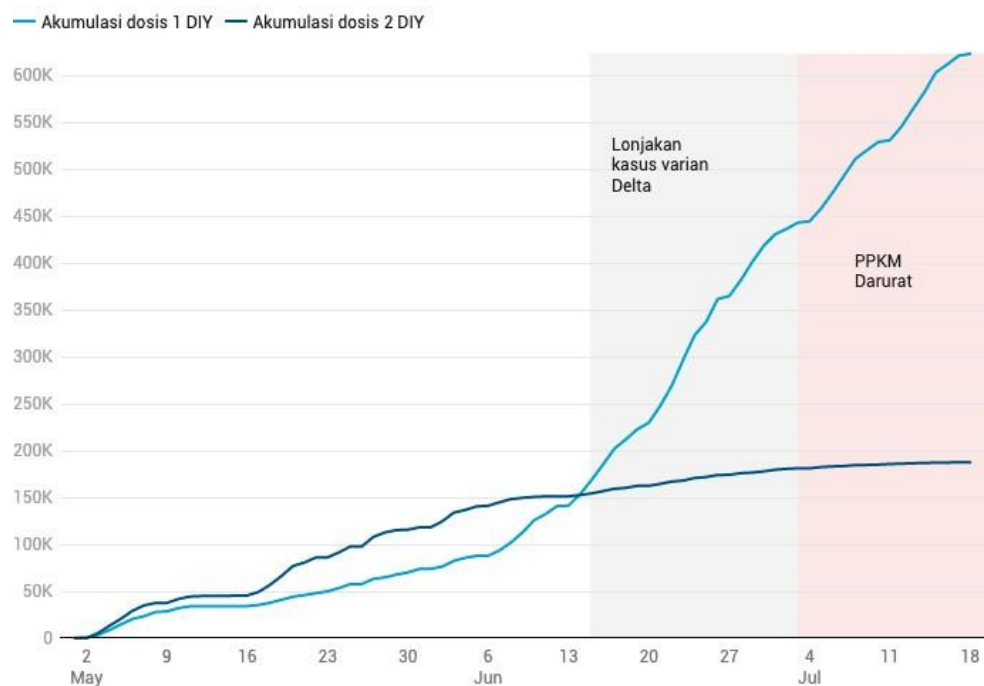
Data vaksinasi yang telah diperoleh diolah untuk dijadikan visualisasi grafik menggunakan Datawrapper. Pada data ditambahkan baris baru yaitu akumulasi vaksinasi untuk dapat melihat akumulasi masyarakat yang telah melakukan vaksinasi di setiap kabupaten/kota di Yogyakarta. Akumulasi vaksinasi ditunjukkan sebagaimana di Gambar 3.22.

Akumulasi Bantul	Akumulasi Gunungkidul	Akumulasi Kulon Progo	Akumulasi Sleman	Akumulasi Yogyakarta
111	0	42	56	0
111	0	42	56	1
1437	445	116	2546	817
5658	830	698	4751	1315
7731	1364	1383	7237	3008
9198	1979	2244	10039	5673
9961	2182	2672	12002	8098
10962	2243	3057	12774	8551
10962	2243	3057	12774	8551
12386	2657	3223	14436	9085
12577	3018	3618	15593	9636
12577	3067	3707	15782	9830
12577	3067	3707	15783	9830
12577	3067	3707	15783	9830
12668	3284	3707	15783	10071
12668	3284	3707	15783	10071
13792	3733	3883	16868	11105
14868	5296	4748	19863	12165
15843	6433	5552	23836	14778
16937	7756	6609	28530	17200
18098	8129	7104	29653	17780
18424	8763	7608	31726	19625
18433	8763	7608	31727	19627
19781	9512	7649	34336	20302
20981	10418	8376	36733	21349
20981	10436	8376	36813	21349
23986	11601	9412	40228	22815
26899	11961	9973	40574	23414
28409	11996	10542	40587	23791
28414	11996	10542	40595	24179
29132	12525	10744	41226	24651

Gambar 3.22 Data Akumulasi Vaksin Dosis Kedua

Hasil visualisasi grafik ditunjukkan pada Gambar 3.23. Berdasarkan hasil tersebut, akumulasi masyarakat DIY yang melakukan vaksinasi terus meningkat. Dengan adanya grafik tersebut dapat memudahkan dalam melihat antusiasme masyarakat dalam melakukan vaksinasi.

### Akumulasi Vaksin DIY



Gambar 3.23 Grafik Vaksinasi

### 3.3.5 Report Sentimen Publik dan Hoaks Selama Pandemi

Tujuan *task* ini diberikan untuk melihat sentimen publik dan hoaks yang ada di media selama masa pandemi. Pengerjaan *task* ini dengan mencari sentimen-sentimen publik di media terkait kata kunci mengenai bantuan sosial, covid, vaksin, isolasi terpadu, Sri Sultan, PPKM, dan Malioboro dalam rentang waktu satu minggu. *Task* ini diberikan mingguan. Beban pengerjaan *task* dibagi secara adil. *Report* sentimen publik dan hoaks ini berbentuk *slide power point*. Untuk memudahkan dalam pengerjaan digunakan Google Slides.

Analisis dilakukan dengan mengolah data *crawling* media sosial Twitter mengenai kata kunci terkait yang diberikan oleh mentor *Data Analyst* Tim Jogja Center. Olah data dilakukan menggunakan *tool* Google Colab dan bahasa pemrograman Python sebagaimana yang ditunjukkan di Gambar 3.24.

```

1 df = pd.read_csv('twitter_bansos.csv', sep=";")
2 df["sentiment"].value_counts()
3 df["type"].value_counts()
4 df["mentions"].value_counts().head(5)

```

Gambar 3.24 Kode Program

Berdasarkan Gambar 3.24, pada baris ke-2 untuk menghitung jumlah sentimen positif, negatif, dan netral yang terdapat dalam *dataset*. Selanjutnya, baris ke-3 dan ke-4 digunakan untuk menghitung jumlah *tweet*, *retweet*, dan *mention* pada *dataset*. Hasil dari Gambar 3.24 selanjutnya diubah menjadi *pie chart*. Kemudian untuk mengetahui lonjakan *tweets* dilakukan pencarian *tweets timeseries* pada *dataset* berdasarkan tanggal dan jam menggunakan Python seperti yang ditunjukkan di Gambar 3.25.

Hasil dari Gambar 3.25 adalah tabel *timeseries*. Tabel tersebut selanjutnya dibuat visualisasi berbentuk grafik. Untuk memudahkan dalam melihat sentimen-sentimen publik, *pie chart* pada Gambar 3.24 serta grafik pada Gambar 3.25 dijadikan satu *slide* seperti yang ditunjukkan pada Gambar 3.26 dan Gambar 3.27. Di *slide* tersebut ditambahkan *tweet* terkait untuk memberikan fakta berdasarkan *dataset* yang telah diberikan. Selanjutnya, terdapat pemantauan hoaks yang berisi hoaks yang muncul di masyarakat selama kurun waktu seminggu. Berikut contoh dari *slide* pemantauan hoaks yang ditunjukkan pada Gambar 3.28.

```

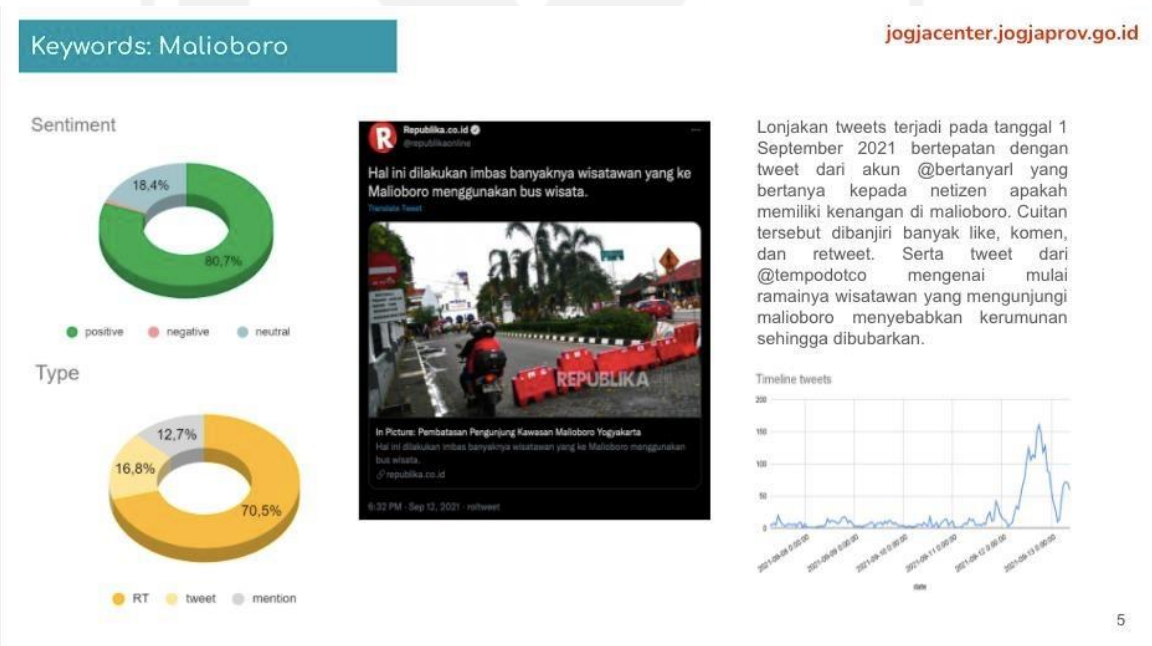
1 #timeseries day
2 df['date'] = pd.to_datetime(df['created']).dt.date
3 df.loc[(df.created == '2021-08-31')] #tanggal timeseries terakhir
4 by_date = pd.Series(df['user.created_at']).value_counts().sort_index()
5 by_date.index = pd.DatetimeIndex(by_date.index)
6 df_date = by_date.rename_axis('date').reset_index(name='counts')
7 df_date.head(30)
8 df_date.to_csv("results/timeseries_date.csv", sep=";", index=False,
9 date_format="%d %B")
10 #timeseries hour
11 df['user.created_at'] = pd.to_datetime(df['created']).dt.strftime('%Y-%m-%d %H:00:00')
12 by_time = pd.Series(df['user.created_at']).value_counts().sort_index()
13 by_time.index = pd.DatetimeIndex(by_time.index)
14 df_time = by_time.rename_axis('time').reset_index(name='counts')
15 df_time.head()
16 df_time.to_csv("results/timeseries_hour.csv", sep=";", index=False)

```

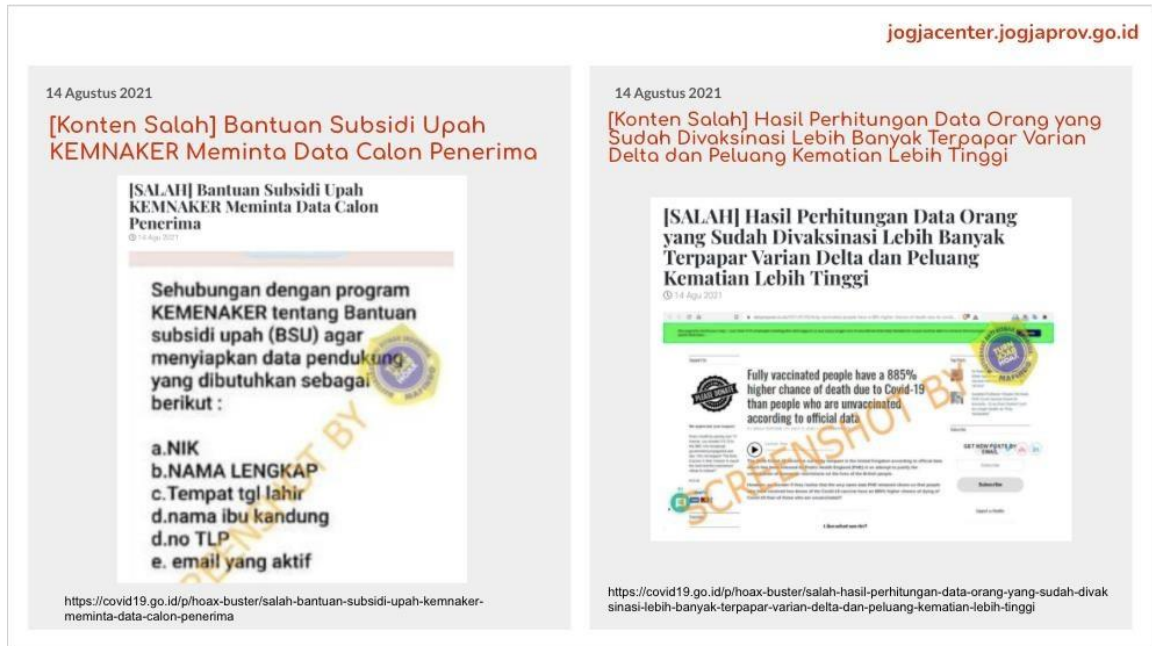
Gambar 3.25 Kode Program *Timeseries*



Gambar 3.26 Sentimen Publik Bansos



Gambar 3.27 Sentimen Publik Malioboro



Gambar 3.28 Pemantauan Hoaks

## BAB IV

### REFLEKSI PELAKSANAAN MAGANG

#### 4.1 Teknis

Sepanjang melakukan magang di Diskominfo DIY sebagai *Data Analyst* diberikan tanggung jawab untuk melakukan mengolah data dan membuat visualisasi dari data yang telah diberikan. Terdapat banyak hal baru yang didapatkan selama kegiatan magang berlangsung. Salah satu *task* yang diberikan adalah analisis sentimen pada ulasan tempat wisata berdasarkan aspek. Dari *task* tersebut diperoleh pengalaman baru seperti cara untuk melakukan analisis sentimen yaitu dengan melakukan pelabelan terlebih dahulu, kemudian dibuat model untuk mengolah data tersebut. Sebelumnya telah didapatkan dasar-dasar mengenai *data science* pada mata kuliah Fundamen Sains Data sehingga membantu dalam pembuatan model *machine learning*. Untuk memilih algoritma yang akan digunakan dilakukan riset terlebih dahulu melalui publikasi yang ada di internet.

Setelah algoritma telah ditentukan, dilakukan pemodelan. Dalam melakukan pemodelan terdapat beberapa langkah yaitu *pre-processing*, klasifikasi, dan evaluasi. Tahap *pre-processing* sangat penting dikarenakan *pre-processing* yang baik akan membantu dalam meningkatkan kualitas model yang akan dibuat. Pada tahapan klasifikasi, terdapat beberapa variabel yang memengaruhi hasil model yaitu pada algoritma Support Vector Machine Model contohnya seperti jenis kernel yang digunakan, nilai *gamma*, dan parameter lainnya. Hasil dari model yang dibuat berupa nilai akurasi dan *confusion matrix*. Penggunaan *confusion matrix* membantu dalam melihat ketepatan model dalam melakukan prediksi pada data uji.

Selanjutnya model yang telah dibuat akan diolah oleh *Data Analyst* lainnya di Tim Jogja Center menjadi *insight data platform online travel* pada dasbor Jogja Center bagian pariwisata. Dasbor ini membantu Pemerintah DIY untuk mendapatkan umpan balik dari pengunjung tempat wisata mengenai tempat-tempat wisata yang ada di DIY. Dasbor *insight data platform online travel* seperti yang ditunjukkan pada Gambar 4.1. Umpan balik tersebut dapat dijadikan sebagai bahan untuk melakukan evaluasi dalam mengembangkan pariwisata di DIY. Refleksi yang didapatkan dari *task* ini adalah seorang *Data Analyst* diminta untuk dapat mengolah data menjadi suatu informasi berguna yang dapat dijadikan menjadi acuan untuk pengambilan suatu keputusan atau solusi dari sebuah permasalahan yang diberikan.



Gambar 4.1 Dasbor *insight data platform online travel*

## 4.2 Non-teknis

Selama magang di Diskominfo DIY, banyak pelajaran non teknis yang didapat. Yakni seperti:

### 4.2.1 Manfaat Magang

Selama 6 (bulan) pelaksanaan magang, banyak pembelajaran yang didapat. Dari sisi *hard skill*, penulis belajar belajar cara melakukan analisis sentimen, melakukan *labelling* yang benar, dan membuat model *machine learning* untuk analisis sentimen. Hal ini belum dipelajari sebelumnya selama masa kuliah. Untuk sisi *soft skill*, kegiatan magang ini sangat membantu untuk meningkatkan kualitas diri, seperti bertanggung jawab dengan tugas yang sudah diberi, cara berkomunikasi yang baik agar tidak terjadi miskomunikasi, dan bersikap profesional dalam dunia kerja.

### 4.2.2 Kendala, Hambatan, dan Tantangan

Hambatan yang dialami selama kegiatan magang berlangsung adalah manajemen waktu. Penulis diharuskan untuk dapat membagi waktu antara kegiatan kuliah dan magang dengan baik guna hindari sesuatu yang tidak diinginkan, seperti tugas yang melewati tenggat waktu maupun tugas yang terabaikan. Karena hal tersebut, dapat melakukan manajemen waktu dengan baik menjadi tantangan yang harus dihadapi oleh penulis. Tantangan lainnya yang dihadapi adalah mempelajari hal-hal yang belum pernah dipelajari dengan cepat untuk dapat mengerjakan *tasks*

yang telah diberikan. Hal ini dapat dilakukan dengan meningkatkan literasi membaca, terutama di waktu luang.

#### 4.2.3 Kontribusi Selama Magang

Selama melaksanakan magang di Diskominfo DIY, penulis terlibat dalam pengembangan web Jogja Center. Terdapat beberapa *task* yang dikerjakan yaitu, lakukan analisis sentimen sesuai aspek di data ulasan tempat wisata di Yogyakarta dan membantu dalam perbaikan fitur sentimen negatif media sosial Twitter dengan melakukan *labelling* secara manual. Adapun *task* yang dikerjakan bersama peserta magang lainnya yaitu, membuat list data tempat wisata di Yogyakarta, list data putusan pengadilan di Yogyakarta, *report* korelasi pergerakan antara Google Mobility dan kasus COVID-19 di Yogyakarta, *report* akumulasi vaksinasi masyarakat DIY, dan *report* sentimen publik dan hoaks selama pandemi.





## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan analisis sentimen pada ulasan tempat wisata berdasarkan aspek yang telah dilakukan menggunakan algoritma Multinomial Naïve Bayes dan Support Vector Machine didapatkan kesimpulan yaitu:

- a. Hasil dari pelabelan 1922 data ulasan pengguna secara manual adalah untuk aspek *attraction* dengan 1313 data positif, 75 data netral, 24 data negatif, dan 510 data bukan kategori. Aspek *accessible* dengan 161 data positif, 21 data netral, 45 data negatif, dan 1695 data bukan kategori. Aspek *amenities* dengan 334 data positif, 9 data netral, 229 data negatif, dan 1350 data bukan kategori. Aspek *ancillary services* dengan 161 data positif, 8 data netral, 30 data negatif, dan 1723 data bukan kategori.
- b. Akurasi yang dihasilkan oleh algoritma Support Vector Machine lebih baik dibanding dengan algoritma Multinomial Naïve Bayes Algoritma SVM menghasilkan nilai 66.49% pada aspek *attraction*, 87.27% pada aspek *accessible*, 69.35% pada aspek *amenities*, dan 90.39% pada aspek *ancillary services*.
- c. *F1-Score* yang dihasilkan oleh algoritma Support Vector Machine untuk sentimen *neutral* menunjukkan nilai yang tidak lebih baik daripada penggunaan metode Multinomial Naïve Bayes.

#### 5.2 Saran

Ada beberapa saran yang diberikan selama melakukan kegiatan magang di Dinas Komunikasi dan Informatika DIY:

- a. Analisis Sentimen Tempat Wisata Berdasarkan Aspek
  - o Untuk mencari algoritma yang terbaik dalam melakukan klasifikasi dapat dengan mencoba menggunakan algoritma lainnya sehingga dapat dijadikan sebagai pembandingan dari model yang telah dibuat.
  - o Menggunakan *dataset* yang seimbang sehingga model dapat melakukan prediksi dengan baik.
- b. Mahasiswa yang akan mengambil jalur magang sebaiknya memilih bidang yang diminati. Tujuannya supaya mahasiswa tersebut dapat mengembangkan diri dengan melakukan

*explore* selama magang dan memberi kontribusi untuk setiap *task* yang diberikan. Apabila memilih bidang yang tidak diminati akan memberikan kesulitan selama magang berlangsung dikarenakan tidak memiliki keinginan untuk mempelajari mengenai bidang tersebut sehingga menghambat dalam pengerjaan tugas.



## DAFTAR PUSTAKA

- Cooper, C., Fletcher, J., David, Gilbert, & Wanhill, S. (1994). *Tourism: Principles and Practice*. Chris Cooper, John Fletcher, David, Gilbert, and Stephen Wanhill. Trans-Atlantic Publications, 311 Bainbridge Street, Philadelphia, PA 19147. 1993. 290p. \$47.50. *Journal of Travel Research*, 32(3), 72–73. <https://doi.org/10.1177/004728759403200326>
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>
- Farisi, A. A., Sibaroni, Y., & Faraby, S. al. (2019). Sentiment analysis on hotel reviews using Multinomial Naïve Bayes classifier. *Journal of Physics: Conference Series*, 1192, 12024. <https://doi.org/10.1088/1742-6596/1192/1/012024>
- Fitriawan, A., Kusuma, W. A., & Heryanto, R. (2013). A classification system for Jamu efficacy based on formula using Support Vector Machine. *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 291–295. <https://doi.org/10.1109/ICACSIS.2013.6761591>
- Goel, A., Gautam, J., & Kumar, S. (2016). Real time sentiment analysis of tweets using Naive Bayes. *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, 257–261. <https://doi.org/10.1109/NGCT.2016.7877424>
- Haykin, S. S. (1999). *Neural networks : a comprehensive foundation*. Prentice Hall.
- Huang, Y., & Li, L. (2011). Naive Bayes classification algorithm based on small sample set. *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, 34–39. <https://doi.org/10.1109/CCIS.2011.6045027>
- Liesnawan, K. I. (2019). *Analisis Sentimen Berdasarkan Aspek Menggunakan Bayesian Network*. <http://elibrary.unikom.ac.id/id/eprint/1479>
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
- Ruuska, S., Hämäläinen, W., Kajava, S., Mughal, M., Matilainen, P., & Mononen, J. (2018). Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle. *Behavioural Processes*, 148, 56–62. <https://doi.org/10.1016/j.beproc.2018.01.004>
- Santosa, B. (2007). *Tutorial Support Vector Machines*.
- Septianti, W. P. (2020). *Analisis Sentimen Berdasarkan Aspek Menggunakan Learning Vector Quantization*. <http://elibrary.unikom.ac.id/id/eprint/2746>

- Siregar, O. P. (2020). *Analisis Sentimen Berdasarkan Aspek Menggunakan Relevance Vector Machine*. <http://elibrary.unikom.ac.id/id/eprint/4116>
- Witten, I., Frank, E., & Hall, M. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). Elsevier. <https://doi.org/10.1016/C2009-0-19715-5>
- Yanti, W. N. (2018). *Analisis sentimen media sosial (Twitter) terhadap layanan provider telekomunikasi (Telkomsel) menggunakan metode multinomial Naïve Bayes* [Doctoral Dissertation, Universitas Islam Negeri Maulana Malik Ibrahim]. <http://etheses.uin-malang.ac.id/id/eprint/12287>



## 1. LAMPIRAN

### A. Daftar *Stop words* (Bahasa Indonesia)

aa	com	gt
aah	dah	gue
ada	dahulu	guna
adalah	dalam	gw
agak	dan	hal
agar	dapat	hanya
ah	dari	harus
aja	daripada	ia
akan	demi	ingin
aku	demikian	ini
amat	dengan	itu
anda	di	itulah
antara	dia	iya
anu	dimana	jadi
apa	dll	jd
apakah	dlm	jika
apalagi	doang	juga
atau	doank	kah
bagaimana	dong	kalau
bagaimanapun	donk	kali
bagi	dsb	kalo
bahwa	dst	kami
begitu	dua	kamu
belum	dulu	kan
bgmn	dulunya	karena
bgt	enggak	kau
biar	ga	ke
bisa	gak	kecuali
boleh	gimana	kemana
buat	gitu	kembali

kenapa	nggak	selagi
kepada	ntar	selain
ketika	nya	sementara
kini	oh	seolah
kita	ok	seperti
km	oleh	sepertinya
knp	pada	seraya
kok	para	serta
koq	pas	sesuatu
kpd	pasti	sesudah
ku	pic	setelah
lagi	pst	seterusnya
lah	pula	setiap
lain	pun	setidaknya
langsung	rt	shg
lg	saat	si
lihat	saja	sih
lo	sama	sj
lu	sambil	spt
maka	sampai	sptny
mana	saya	spy
mari	sblm	sudah
masih	scr	supaya
masuk	sdg	sy
mau	sebab	tanpa
melainkan	sebagai	tapi
mengapa	sebelum	tau
menurut	sebetulnya	tdk
mereka	secara	telah
mu	sedangkan	tentang
namun	seharusnya	tentu
nanti	sehingga	terhadap
ngga	sekitar	terjemahan

tetapi	untuk	yang
tidak	utk	yeah
to	walau	yes
toh	ya	yg
tolong	yaa	yha
tp	yah	yo
udah	yaitu	
udh	yakni	

B. Daftar *Stop words* (Bahasa Inggris)

see	herself	been
let	it	being
u	its	have
i	itself	has
me	they	had
my	them	having
myself	their	do
we	theirs	does
our	themselves	did
ours	what	doing
ourselves	which	a
you	who	an
your	whom	the
yours	this	and
yourself	that	but
yourselves	these	if
he	those	or
him	am	because
his	is	as
himself	are	until
she	was	while
her	were	of
hers	be	at

by	all
for	any
with	both
about	each
against	few
between	more
into	most
through	other
during	some
before	such
after	no
above	nor
below	not
to	only
from	own
up	same
down	so
in	than
out	too
on	very
off	s
over	t
under	can
again	will
further	just
then	don
once	should
here	now
there	
when	
where	
why	
how	

