

**ANALISIS SENTIMEN TWITTER TERHADAP PROFESI TUKANG
PARKIR MENGGUNAKAN METODE *NAÏVE BAYES***

Skripsi
untuk memenuhi sebagai persyaratan
mencapai derajat Sarjana S-1
Program Studi Teknik Informatika



Disusun oleh:
Muhammad Sholihuddin Nur
16650044

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA
2020**

HALAMAN PENGESAHAN

SURAT PERSETUJUAN SKRIPSI

PERNYATAAN KEASLIAN SKRIPSI

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penelitian yang dilakukan dengan judul **Analisis Sentimen Twitter Terhadap Profesi Tukang Parkir Menggunakan Metode *Naïve Bayes*** dapat berjalan dengan baik. Sholawat serta salam semoga tetap tercurahkan kepada junjungan kita Nabi Muhammad SAW.

Tidak lupa penulis juga mengucapkan terima kasih kepada seluruh pihak yang telah membantu dalam penyelesaian skripsi ini. Ucapan terimakasih penulis sampaikan kepada:

1. Kedua orang tua, Bapak Sutarman S.E dan Ibu Lilis Yuliatun yang telah memberikan biaya, dukungan, semangat dan do'a sehingga penulis dapat selalu berjuang dalam hidup dan berusaha menjadi pribadi yang bermanfaat bagi orang lain.
2. Bapak Muhammad Didik Rohmad Wahyudi, S.T., MT. Selaku dosen pembimbing skripsi yang telah sabar membimbing, memberikan arahan, dan motivasi dalam penyusunan skripsi.
3. Bapak Dr. Agung Fatwanto, S.Si., M.Kom., selaku Dosen Pembimbing Akademik.
4. Seluruh dosen pengampu mata kuliah program studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta.
5. Sahabatku yang tidak perlu disebut namanya yang telah menemani dan membersamai pendewasaan hidup penulis.

6. Sahabat-sahabat Pagardepan Dagadu Djokdja yang telah memberikan warna dalam kehidupan penulis.
7. Seluruh teman-teman Teknik Informatika angkatan 2016 yang tidak dapat penulis sebutkan satu per satu.
8. Teman-teman IMAGIRI Yogyakarta selalu menemani dan membuat penulis semangat dalam menyelesaikan tugas akhir.
9. Teman-teman KKN dusun Sumberjo yang telah menjadi pengalaman mengesankan dalam belajar bermasyarakat.
10. Serta seluruh pihak yang tidak dapat disebutkan satu persatu dan telah memberikan banyak doa dan dukungan sehingga penelitian ini dapat terselesaikan.

Yogyakarta, 25 Maret 2020

Penulis

HALAMAN PERSEMBAHAN

Skripsi ini penulis persembahkan kepada kedua orang tua, yang memberi kebebasan dengan pemahaman.

Sehingga penulis dapat belajar banyak hal terlepas dari pendidikan yang ada.

Dan kepada diriku sendiri,

.

HALAMAN MOTTO

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya”

- Quran Surat Al-Baqarah Ayat 286

“Sebaik-baiknya manusia adalah yang bermanfaat bagi orang lain”

- Hadist R. Ahmad

“If I die tomorrow i'd be all right, because I believe.

That after we're gone The spirit carries on.”

-Dream Theater-

DAFTAR ISI

HALAMAN PENGESAHAN.....	II
SURAT PERSETUJUAN SKRIPSI.....	III
PERNYATAAN KEASLIAN SKRIPSI.....	IV
KATA PENGANTAR.....	V
HALAMAN PERSEMBAHAN	VII
HALAMAN MOTTO	VIII
DAFTAR ISI.....	IX
DAFTAR GAMBAR.....	XIII
DAFTAR TABEL	XIV
INTISARI	XVII
ABSTRACT	XVIII
BAB I.....	1
PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3

1.5.	Manfaat Penelitian.....	3
1.6.	Keaslian Penelitian	4
1.7.	Sistematika Penulisan.....	4
BAB II	6
TINJAUAN PUSTAKA DAN LANDASAN TEORI	6
2.1.	Tinjauan Pustaka	6
2.2.	Landasan Teori	11
2.2.1.	<i>Teks Mining</i>	11
2.2.2.	<i>Text Preprocessing</i>	12
2.2.3.	<i>Analisis Sentimen</i>	13
2.2.4.	<i>Python</i>	14
2.2.5.	<i>Twitter</i>	14
2.2.6.	<i>Klasifikasi</i>	14
2.2.7.	<i>Naïve Bayes</i>	14
2.2.8.	<i>TF-IDF</i>	16
2.2.9.	<i>Split Validation</i>	17
2.2.10.	<i>Confusion Matrix</i>	17
2.2.11.	Tukang Parkir	18
BAB III	19
METODE PENELITIAN	19
3.1.	Metode Penelitian.....	19
3.2.	Alur Penelitian.....	19

3.2.1.	Studi Pustaka.....	19
3.2.2.	Pengumpulan Data	19
3.2.3.	Seleksi Data.....	20
3.2.4.	<i>Preprocessing</i> Data	20
3.2.5.	Analisa.....	20
3.2.6.	Implementasi	21
3.2.7.	Pembuatan Laporan.....	21
3.3.	Perangkat Penelitian	21
3.3.1.	Perangkat Keras	21
3.3.2.	Perangkat Lunak.....	21
BAB IV	23
HASIL DAN PEMBAHASAN	23
4.1.	Pengumpulan Data	23
4.2.	Pelabelan dan Seleksi Data.....	24
4.3.	<i>Preprocessing</i> Data	25
4.3.1.	Cleansing.....	25
4.3.2.	Case Folding	26
4.3.3.	Slangword Removal	27
4.3.4.	Stopword Removal.....	28
4.3.5.	Stemming	29
4.4.	Pembobotan Kata (Term Weighting)	30
4.4.1.	Term Frequency – Invers Document Frequency (<i>TF-IDF</i>)	31
4.5.	Algoritma Naïve Bayes	34

4.6. Evaluasi Model Klasifikasi	39
4.7. Implementasi	43
BAB V.....	46
PENUTUP.....	46
5.1. Kesimpulan	46
5.2. Saran.....	47
DAFTAR PUSTAKA	48
LAMPIRAN.....	50
CURRICULUM VITAE.....	61

DAFTAR GAMBAR

Gambar 4.1 Flowchart Proses Bisnis	23
Gambar 4.2 Grafik Confusion Matrix Percobaan Pertama	41
Gambar 4.3 Grafik Confusion Matrix Percobaan Kedua.....	42
Gambar 4.4 Grafik Confusion Matrix Percobaan Ketiga.....	42
Gambar 4.5 Grafik Hasil Implementasi	44
Gambar 4.6 Visualisasi prediksi kelas positif dengan Wordcloud.....	44
Gambar 4.7 Visualisasi prediksi kelas negatif dengan Wordcloud.....	45

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka	9
Tabel 2.2 Tinjauan Pustaka (lanjutan)	10
Tabel 2.3 Tinjauan Pustaka (lanjutan).....	11
Tabel 2.4 Tabel Confusion Matrix	17
Tabel 4.1 Contoh Data Crawling.....	24
Tabel 4.2 Contoh Data dengan Label	25
Tabel 4.3 Contoh Data Cleansing.....	26
Tabel 4.4 Contoh Data dengan Casefolding.....	27
Tabel 4.5 Contoh Data dengan Slangword Removal	28
Tabel 4.6 Contoh Data dengan Stopword Removal	29
Tabel 4.7 Contoh Data dengan Stemming.....	29
Tabel 4.8 Contoh Data dengan Stemming (lanjutan)	30
Tabel 4.9 Contoh Data untuk Pembobotan	30
Tabel 4.10 Contoh Perhitungan IDF	31
Tabel 4.11 Contoh Perhitungan IDF (lanjutan).....	32
Tabel 4.12 Contoh Perhitungan TF-IDF	32
Tabel 4.13 Contoh Perhitungan TF-IDF (lanjutan).....	33
Tabel 4.14 Contoh Perhitungan TF-IDF (lanjutan).....	35
Tabel 4.15 Contoh Perhitungan Probabilitas Laplace Smoothing.....	35
Tabel 4.16 Contoh Perhitungan Probabilitas Laplace Smoothing (lanjutan)	36
Tabel 4.17 Contoh Perhitungan Kelas Negatif.....	38

Tabel 4.18 Contoh Perhitungan Kelas Positif	38
Tabel 4.19 Contoh Perhitungan Kelas Positif (lanjutan).....	39
Tabel 4.20 Skenario Pengujian Split Validation	40
Tabel 4.21 Hasil Pengujian Split Validation	43

DAFTAR RUMUS

Probabilitas (1).....	15
Prior Probability (2)	15
Conditional Probability (3)	15
Conditional Probability dengan Laplace Smoothing (4).....	16
TF-IDF (5).....	16
Menghitung Akurasi (6).....	18
IDF (7).....	31
Menghitung probabilitas kata ke -n (8).....	34
Menghitung probabilitas dokumen kelas c (9).....	37

Analisis Sentimen Twitter Terhadap Profesi Tukang Parkir

Menggunakan Metode *Naïve Bayes*

Muhammad Sholihuddin Nur

16650044

INTISARI

Keberadaan profesi tukang parkir menjadi sebuah bahasan yang bisa cukup menguras energi bila diperbincangkan, baik itu segi fungsionalitas, nilai dan harga jasa yang diberikan, dan juga urgensi keberadaanya, karena belum ada penelitian yang membahas tentang hal tersebut. Maka dari itu, dilakukan penelitian mengenai Analisis Sentimen terhadap Profesi Tukang Parkir.

Penelitian ini melakukan klasifikasi terhadap data yang diambil dari twitter sejumlah 1357 data untuk dianalisa dengan metode *Naïve Bayes* dengan *TF-IDF* sebagai model pembobotan kata (term weighting), serta menggunakan *split validation* dan *confusion matrix* sebagai tahap evaluasi model, lalu selanjutnya dilakukan implementasi terhadap 1000 data baru untuk dilakukan prediksi.

Rasio *split validation* yang digunakan pada masing-masing model menunjukkan akurasi yang berbeda dengan menggunakan evaluasi *confusion matrix* model dengan rasio split 90:10 menghasilkan akurasi 82,35%, rasio split 80:20 menghasilkan akurasi 79,04%, sedangkan rasio split 70:30 menghasilkan akurasi 78,92%.

Hasil implementasi pada data uji sejumlah 1000 tweet yang diambil pada rentang waktu 7-12 Juli menggunakan model yang sudah dibuat menghasilkan prediksi tweet positif sebanyak 28,9% dan prediksi tweet negatif sebanyak 71,1%.

Kata kunci: *analisis sentimen, twitter, naïve bayes, TF-IDF, split validation, confusion matrix, tukang parkir*

Sentiment Analysis Towards Profession of Parkingman

Using *Naïve Bayes* Classifier

Muhammad Sholihuddin Nur

16650044

ABSTRACT

The existence of the parking attendant profession is a topic that can be quite energy-consuming when discussed, both in terms of functionality, value and price of services provided, and also the urgency of its existence, because there is no research that discusses it. Therefore, a study of sentiment analysts on the parking attendant profession is carried out.

This study classifies 1357 data taken from Twitter to be analyzed by the *Naïve Bayes* method with TF-IDF as a term weighting model, and uses a split validation and confusion matrix as the evaluation stage of the model, then the next implementation of 1000 new data to predict.

The split validation ratio used in each model shows different accuracy using the evaluation of the confusion matrix. Model with a split ratio of 90:10 producing an accuracy of 82.35%, a split ratio of 80:20 producing an accuracy of 79.04%, while a split ratio of 70:30 resulting in an accuracy of 78.92%.

The results of the implementation to the test data of 1000 tweets taken in the span of July 7-12 using the model produces positif tweet predictions of 28.9% and negatif tweet predictions of 71.1%.

Keywords: sentiment analysis, twitter, naïve bayes, TF-IDF, split validation, confusion matrix, parking attendant

BAB I

PENDAHULUAN

1.1. Latar Belakang

Keberadaan profesi tukang parkir menjadi sebuah bahasan yang bisa cukup menguras energi bila diperbincangkan, baik itu segi fungsionalitas, nilai dan harga jasa yang diberikan, dan juga urgensi keberadaannya. Menurut situs manajemen parkir GBParking, pekerjaan tukang parkir muncul secara alamiah sebelum tahun 1955 di Jakarta dari penduduk setempat tanpa ada kebijakan maupun undang-undang yang mengatur pekerjaan tukang parkir. Proyek Asean Games pada tahun 1962 turut andil menjadi sebab menjamurnya profesi tukang parkir di sekitar tempat perhelatan Asean Games. Dikarenakan saat itu bisnis parkir mulai diwarnai oleh banyaknya kasus bentrok perebutan lahan pemerintah daerah dalam hal ini Dinas Pekerjaan Umum (DPU) mulai ikut andil dalam mengatur bisnis perparkiran di Jakarta. Hingga saat ini sudah banyak kebijakan yang dikeluarkan oleh pemerintah otonom yang mengatur tentang usaha parkir di masing-masing pemerintahan daerah, salah satunya di PERDA Kota Yogyakarta nomor 2 tahun 2019 yang mengatur tentang usaha parkir di Yogyakarta. Namun di lapangan hingga saat ini masih banyak oknum tukang parkir, baik itu individu maupun kelompok, yang melakukan usaha parkir dengan ilegal. Dewasa ini, profesi tukang parkir baik itu tukang parkir resmi maupun liar masih menjadi polemik di masyarakat terkait dengan fungsi, nilai jasa, dan keberadaannya. Banyak sekali pro dan kontra terkait keberadaan tukang parkir, namun sampai saat ini masih hanya berupa asumsi dan

opini. Platform Twitter adalah salah satu media sosial yang digemari masyarakat. Twitter menyediakan kemudahan dalam mengirim atau menerima informasi dengan singkat dan cepat, dari kemudahan dan kecepatan tersebut menjadi salah satu alasan terus meningkatnya jumlah pengguna twitter setiap hari-nya. Pada laporan finansial Twitter kuartal ke-3 tahun 2019, pengguna aktif harian di platform Twitter Global dicatat meningkat 17 persen, ke angka 145 juta pengguna, dan laman kominfo mencatut berita bahwa Indonesia menempati peringkat ke-5 pengguna twitter terbanyak, karena itu Twitter dapat menjadi sumber data pendapat yang bisa digunakan secara efisien untuk pemasaran dan studi sosial.

Berdasarkan data tweet sejumlah 5017 data yang diambil pada bulan November 2019 dengan pencarian kata kunci tukang parkir. Penulis dalam penelitian ini akan melakukan analisa tentang opini tweet dengan membuat model pembelajaran mesin yang diharapkan bisa mengklasifikasikan opini tweet mengenai profesi tukang parkir di Indonesia menggunakan metode *Naïve Bayes Classifier* dengan baik. Dengan adanya penelitian, ini diharapkan dapat memberikan wawasan baru yang berupa hasil penelitian dan selanjutnya bisa digunakan pihak-pihak terkait yang membutuhkan analisis informasi mengenai profesi tukang parkir dalam platform twitter baik itu digunakan untuk penelitian ataupun kebijakan.

1.2. Rumusan Masalah

Berdasar latar belakang di atas, masalah penelitian ini adalah “Bagaimana analisis sentimen tweet terhadap profesi tukang parkir”.

1.3. Batasan Masalah

Dalam penelitian ini terdapat beberapa batasan masalah yang dibahas agar penyusunan dan pembahasan penelitian dapat dilakukan secara terarah dan tercapai sesuai dengan yang diharapkan. Antara lain sebagai berikut :

1. Penelitian ini hanya membahas klasifikasi sentimen masyarakat terhadap profesi tukang parkir.
2. Klasifikasi tweet, mencakup sentimen positif dan negatif
3. Klasifikasi dibuat dengan metode pemodelan *Naïve Bayes*
4. Data yang digunakan sejumlah 5017 data dengan kata kunci ‘tukang parkir’
5. Proses klasifikasi menggunakan bahasa pemrograman python

1.4. Tujuan Penelitian

Mengklasifikasikan sentimen tweet terhadap profesi tukang parkir dengan menerapkan metode *Naïve Bayes* dalam proses pembuatan model klasifikasi dan melihat sejauh mana tingkat akurasi pemodelan menggunakan *Confusion Matrix*.

1.5. Manfaat Penelitian

Adapun manfaat penelitian ini diantaranya :

1. Memberikan wawasan berupa hasil penelitian yang bisa digunakan sebagai sumber pembelajaran atau sebagai acuan untuk mengambil kebijakan.

2. Menambah wawasan dalam implementasi algoritma *Naïve Bayes* dalam pengklasifikasi sentimen twitter terhadap profesi tukang parkir

1.6. Keaslian Penelitian

Penelitian mengenai analisis sentimen dengan metode klasifikasi *Naïve Bayes Classifier*, sampai saat ini sudah dilakukan oleh beberapa peneliti sebelumnya. Namun, berdasarkan referensi dan tinjauan pustaka, penelitian Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta mengenai analisis sentimen menggunakan sosial media twitter dengan metode *Naïve Bayes Classifier* mengenai pandangan masyarakat terhadap profesi tukang parkir belum pernah ditemukan oleh peneliti.

1.7. Sistematika Penulisan

Sebagai gambaran dan kerangka yang jelas mengenai pokok bahasan setiap bab dalam penelitian ini, maka diperlukan sistematika penulisan. Berikut adalah penjelasan pada tiap-tiap bab dalam laporan penelitian ini:

BAB I PENDAHULUAN

Bab pendahuluan berisikan penjelasan mengenai latar belakang dilakukannya penelitian, rumusan masalah penelitian, batasan masalah, tujuan penelitian, manfaat penelitian, keaslian penelitian, dan sistematika penulisan penelitian

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab tinjauan pustaka dan landasan teori berisikan mengenai tinjauan pustaka dan landasan teori yang berhubungan dengan topik yang akan dibahas dalam penelitian ini.

BAB III METODE PENELITIAN

Bab metode penelitian berisi tentang metode ataupun algoritma yang digunakan serta tahapan-tahapan yang akan dilakukan dalam penelitian ini.

BAB IV HASIL DAN PEMBAHASAN

Bab hasil dan pembahasan berisi tentang proses penelitian sampai hasil hasil dari penelitian yang telah dilakukan.

BAB V PENUTUP

Bab penutup menjelaskan tentang kesimpulan dari hasil penelitian yang telah dilakukan. Kekurangan yang ada pada penelitian ini dituliskan pada saran untuk sehingga dapat berguna untuk perkembangan penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Pada tahap ini dilakukan studi literatur atau pencarian referensi, pengumpulan data jurnal penelitian maupun skripsi terkait. Hasil penelitian terdahulu yang dilakukan oleh Mahardika dan Zuliarso (2018) “Analisis Sentimen Terhadap Pemerintahan Joko Widodo Pada Media Sosial Twitter Menggunakan Algoritma *Naïve Bayes Classifier* menggunakan data 400 data tweet yang terdiri dari 300 data latih dan 100 data uji, untuk data latih sendiri terdiri dari 150 tweet dengan sentimen negatif dan 150 sentimen positif. Tahap *preprocessing* terdiri dari *case folding*, normalisasi fitur, *tokenizing* dan *stopword removal*. Proses klasifikasi menggunakan *Naïve Bayes Classifier*, lalu untuk menghitung tingkat akurasi menggunakan *confusion matrix*. Tahap pemodelan menggunakan 300 data dengan rasio pembagian 300 data latih dan 100 data uji. Dalam penelitian ini Metode *Naïve Bayes Classifier* dalam melakukan pemodelan klasifikasi menghasilkan performa akurasi sebesar 97% sehingga dapat dilakukan untuk mengklasifikasi data berikutnya secara otomatis.

Hidayatullah dan Azhari (2014) melakukan penelitian tentang Analisis Sentimen Dan Klasifikasi Kategori Terhadap Tokoh Publik Pada Twitter, Penelitian ini mencoba menganalisis tweet berbahasa Indonesia yang membicarakan tokoh publik tertentu. Sekaligus juga membandingkan antara metode klasifikasi *Naïve bayes* dan *SVM* serta membandingkan fitur *TF-IDF* dan

term frequency dalam setiap metode klasifikasi. Klasifikasi berdasarkan kombinasi kelas sentimen dan kelas kategori, klasifikasi sentimen terdiri dari positif dan negatif sedangkan klasifikasi kategori terdiri dari kapabilitas, integritas, dan akseptabilitas yang diambil dari indikator yang digunakan LSI (Lembaga Survei Indonesia). Proses pelabelan dilakukan secara manual. Proses menghitung akurasi menggunakan aplikasi *Rapid Miner*. Penelitian ini telah berhasil membangun model untuk melakukan klasifikasi tweet berdasarkan kelas sentimen dan kelas kategori dengan *Naïve Bayes Classifier*. Hasil akurasi pengujian klasifikasi *Naïve Bayes* dengan fitur *term frequency* diperoleh sebesar 79,91% sedangkan fitur *TF-IDF* didapatkan akurasi sebesar 79,68%. Klasifikasi menggunakan tools *Rapid Miner* dengan *Naïve Bayes* dan fitur *term frequency* diperoleh sebesar 73,81% sedangkan dengan fitur *TF-IDF* diperoleh sebesar 71,11%. Klasifikasi dengan *Support Vector Machine* menghasilkan akurasi 83,14% untuk fitur *term frequency* dan 82,69% untuk fitur *TF-IDF*. Hasil pengujian pada tools *Rapid Miner* memperlihatkan bahwa penggunaan metode *Support Vector Machine* dan *Naïve Bayes* sama-sama memiliki hasil akurasi yang cukup baik untuk klasifikasi tweet.

Pintoko & L (2018) Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan metode klasifikasi *Naïve Bayes Classifier*, penelitian ini bertujuan melakukan analisis sentimen dari masyarakat terhadap jasa transportasi online diantaranya Gojek dan Grab untuk memberikan informasi tentang kepuasan masyarakat terhadap jasa transportasi online yang ada di Indonesia. Pada penelitian ini untuk tahap crawling data menggunakan tweepy, dengan total 2000 data yang terdiri dari 1500 data latih dan 500 data uji yang dilabeli secara manual. Proses

pemodelan menggunakan metode *Naïve Bayes Classifier* setelah itu melakukan prediksi kelas dengan menggunakan aplikasi *scikit-learns*, data divisualisasikan menggunakan *wordcloud*. Hasil analisis yang dilakukan terkait sentimen terhadap jasa transportasi online dengan menggunakan metode *Naïve Bayes* untuk klasifikasi tweet prediksi, didapatkan hasil sentimen positif sebesar 88.60% dan sentimen negatif sebesar 11.40% dengan akurasi sebesar 86.80%.

Suryono, dkk (2018) Klasifikasi Sentimen Pada Twitter Dengan *Naïve Bayes Classifier*. Penelitian ini membandingkan kinerja *Naïve Bayes Classifier* dengan metode *split validation* dalam melakukan klasifikasi dokumen. Data diambil dari twitter sebanyak 3845 tweet, dengan topik pembahasan presiden Joko Widodo beserta pemerintahannya. Dari uji coba 3 kali dengan rasio *split validation* yang berbeda, pengujian pertama menghasilkan nilai akurasi 64,95% kedua sebesar 66,36% dan pengujian ketiga sebesar 66,79%. Berdasarkan hasil analisis dari pengujian data menggunakan metode dengan akurasi paling tinggi, klasifikasi sentimen positif sebesar 28%, sentimen negatif 20% dan sentimen netral sebesar 52%. Berdasarkan hasil prediksi kelas sentimen, sentimen netral merupakan sentimen yang paling banyak apabila dikaitkan dengan topik Presiden Joko Widodo dan pemerintahannya.

Hamidah (2019) melakukan penelitian tentang analisis sentimen masyarakat terhadap *fintech*, penelitian ini membandingkan dua model pembobotan untuk melihat pengaruh masing masing model pembobotan dalam proses klasifikasi. Dua model yang dipakai adalah *TF* dan *TF-IDF*, selanjutnya akan dilakukan proses klasifikasi dengan metode klasifikasi *Naïve Bayes* , lalu untuk

evaluasi model klasifikasi menggunakan metode *K-Fold Cross Validation* dan *Leave One Out Cross Validation*. Data yang digunakan untuk membangun model klasifikasi berupa data tweet berjumlah 2500 data dengan 4 label (positif, negatif, irrelevant, netral). Hasil dari penelitian ini adalah, model yang dibangun dengan pembobotan *TF* dan diuji menggunakan *K-Fold Cross Validation* menghasilkan akurasi sebesar 76,02% dan 71,68% untuk *Leave One Out Cross Validation*, untuk pembobotan dengan *TF-IDF* menggunakan pengujian *K-Fold Cross Validation* menghasilkan 67,24% akurasi dan untuk *Leave One Out Cross Validation* menghasilkan nilai akurasi 51,80%.

Tabel 2.1 Tinjauan Pustaka

No.	Judul	Peneliti	Metode	Hasil
1	Analisis Sentimen Terhadap Pemerintahan Joko Widodo Pada Media Sosial Twitter Menggunakan Algoritma Naïve Bayes Classifier	Mahardika (2018)	Naïve bayes	Pemodelan Klasifikasi dengan menggunakan metode Naïve Bayes dengan data latih berjumlah 300 terdiri dari 150 label positif dan 150 label negatif menghasilkan model klasifikasi dengan akurasi 97%.
2	Analisis Sentimen Dan Klasifikasi Kategori Terhadap Tokoh Publik Pada Twitter	Hidayatullah (2015)	Naïve Bayes, SVM	Metode Support Vector Machine menghasilkan akurasi performansi yang lebih baik daripada metode Naïve Bayes, dan pembobotan term frequency. Penggunaan fitur term frequency menghasilkan akurasi

Tabel 2.2 Tinjauan Pustaka (lanjutan)

No.	Judul	Peneliti	Metode	Hasil
				yang sedikit lebih baik daripada penggunaan fitur TF-IDF. Penggunaan metode Support Vector Machine dan Naïve Bayes sama-sama memiliki hasil akurasi yang cukup baik untuk klasifikasi tweet.
3	Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan metode klasifikasi Naïve Bayes Classifier	Pintoko & L (2018)	Naïve Bayes	Membuat pemodelan klasifikasi dengan 1500 data latih dengan metode klasifikasi Naïve Bayes, menghasilkan akurasi 86,60%. Lalu model yang sudah dibuat dilakukan implementasi terhadap 500 data latih dan
4	Analisis Sentimen Dan Klasifikasi Kategori Terhadap Tokoh Publik Pada Twitter	Suryono, Utami, Luthfi (2019)	Naïve Bayes, SVM	Metode Support Vector Machine menghasilkan akurasi performansi yang lebih baik daripada metode Naïve Bayes, dan pembobotan term frequency. Penggunaan fitur term frequency menghasilkan akurasi yang sedikit lebih baik daripada penggunaan fitur TF-IDF. Penggunaan metode Support Vector Machine dan Naïve Bayes sama-sama memiliki hasil akurasi yang cukup baik untuk klasifikasi tweet.

Tabel 2.3 Tinjauan Pustaka (lanjutan)

No.	Judul	Peneliti	Metode	Hasil
5	Analisis Sentimen Masyarakat Terhadap Fintech Menggunakan Metode Naïve Bayes Classifier	Hamidah (2019)	Naïve Bayes	Hasil pemodelan menggunakan fitur TF dan pengujian menggunakan K-Fold Cross Validation menghasilkan nilai akurasi yang besar, untuk implementasi dalam penelitian ini menggunakan model yang dibangun menggunakan pembobotan TF-IDF

2.2. Landasan Teori

2.2.1. Text Mining

Text mining adalah proses menganalisa teks untuk mengekstrak informasi dari data yang mempunyai pola yang berguna untuk tujuan tertentu, text mining adalah ilmu yang bertujuan untuk memproses teks agar menjadi informasi yang diperoleh dari peramalan pola dan kecenderungan melalui pola statistik. (Han, Kamber dan Pei, 2012). *Text mining* bertujuan untuk menganalisis pendapat, sentimen, evaluasi, sikap, penilaian, emosi seseorang sehingga dapat diketahui apakah berkenaan dengan suatu topik, layanan, organisasi, individu, atau kegiatan tertentu (Liu, 2012). Penggunaan dari *text mining* dilakukan untuk klasterisasi, klasifikasi, *information retrieval*, dan *information extraction* (Berry & Kogan, 2010).

2.2.2. *Text Preprocessing*

Text preprocessing merupakan salah satu tahap dalam *text mining*, *text preprocessing* merupakan proses mengolah dan mengatur informasi dari data teks yang tidak terstruktur menjadi lebih terstruktur. Tujuan utama teks preprocessing adalah menyederhanakan teks agar dapat dimengerti dan diolah lebih lanjut dalam penelitian (Hamidah, 2019).

Tahapan *text preprocessing* diantaranya sebagai berikut

1. *Cleansing*

Adalah tahap membersihkan data dari karakter-karakter non huruf, seperti *hashtag*, *url*, *username*, tanda baca, ataupun karakter karakter lainnya agar tidak mengganggu proses pengolahan dokumen

2. *Case Folding*

Merupakan tahap menyeragamkan seluruh teks menjadi huruf kecil (*lowercase*), hanya huruf a sampai z yang dapat diterima, karakter selain huruf dihilangkan dan dianggap delimiter.

3. *Tokenizing*

Tokenizing adalah tahap memecah kalimat menjadi token agar nantinya dapat diolah di tahapan tahapan selanjutnya

4. *Slangword Removal*

Slangword removal merupakan tahap untuk mengonversi ragam bahasa musiman tidak baku (*slang*) menjadi kata baku. contoh kata *slang* dalam penggunaan bahasa Indonesia diantaranya yg dengan bentuk baku yang, bgmn dengan bentuk baku bagaimana, jd dengan bentuk baku jadi

5. *Stopword Removal*

Stopword removal merupakan tahap untuk menghilangkan kata yang sering muncul. Yang termasuk dalam *stopword* adalah kata hubung, kata ganti orang, penunjuk, waktu, dan kata tanya.

Dalam proses *preprocessing* penggunaan *stopword removal* dapat meningkatkan performa klasifikasi dengan bukti peningkatan akurasi (Hidayatullah 2016).

6. *Stemming*

Merupakan suatu proses untuk memecah suatu varian-varian kata menjadi bentuk kata dasar (*stem*) dengan menghilangkan imbuhan imbuhan baik itu berupa *prefiks*, *sufiks*, maupun *konfiks* yang ada pada setiap kata.

2.2.3. Analisis Sentimen

Analisis sentimen juga dapat dikatakan sebagai *opinion mining*. Analisis sentimen dapat digunakan dalam berbagai kemungkinan domain, dari produk konsumen, jasa kesehatan, jasa keuangan, peristiwa sosial dan politik pada pemilu. Kecendrungan penelitian tentang analisis sentimen berfokus pada pendapat yang menyatakan atau menyiratkan suatu sentimen positif atau negatif. Pendapat mewakili hampir semua aktivitas manusia, karena pendapat dapat mempengaruhi terhadap perilaku seseorang. Setiap kali kita perlu membuat keputusan, kita ingin tahu pendapat orang lain. Dalam dunia nyata, bisnis dan organisasi selalu ingin melihat opini publik tentang suatu produk atau jasa (Liu, 2012).

2.2.4. *Python*

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar.

2.2.5. *Twitter*

Twitter adalah sebuah layanan jejaring sosial (media sosial) dan juga microblog yang memungkinkan penggunanya berkirim dan membaca pesan yang tidak lebih dari 280 karakter yang disebut sebagai tweet. Sebelumnya, pesan di Twitter hanya sampai 140 karakter tetapi pada tanggal 7 November 2017 ditambah menjadi 280 karakter. Twitter didirikan pada 21 Maret 2006 oleh Jack Dorsey, Noah Glass, Biz Stone, dan Evan Williams. Sosial Media Twitter sendiri dirilis ke publik pada 15 Juli 2006. Maskarnya berada di San Francisco, California, Amerika Serikat.

2.2.6. *Klasifikasi*

Klasifikasi menurut Kursini (2009) adalah proses penemuan model (atau fungsi) yang menggambarkan dan membedakan kelas data atau konsep yang bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui.

2.2.7. *Naïve Bayes*

Naïve Bayes Classifier merupakan salah satu algoritma dalam teknik *text mining* yang menerapkan teori Bayes dalam klasifikasi. Konsep dasar yang

digunakan pada metode *Naïve Bayes* yaitu melakukan klasifikasi dengan melakukan perhitungan nilai probabilitas $P(c|d)$, yaitu probabilitas dokumen d terhadap kelas c .

Menurut Manning, Raghava, & Schutze, probabilitas dokumen berada di kelas c dihitung dengan rumus dibawah ini.

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

$P(t_k|c)$ adalah conditional probability dari term t_k yang terdapat pada kelas c . $P(c)$ merupakan *prior probability* dari dokumen yang terdapat pada kelas c yang dapat dicari menggunakan persamaan berikut.

$$P_l = \frac{N_l}{N} \quad (2)$$

N_l adalah jumlah dokumen kelas c dalam training. Sedangkan N adalah jumlah seluruh dokumen kelas dalam training. Untuk menghitung nilai conditional probability $P(t|k)$ menggunakan persamaan

$$P(W_k|l) = \frac{W_{lt}}{\sum_{w' \in vw'ct} + B'} \quad (3)$$

W_{lt} adalah nilai bobot TF-IDF pada kata t dalam sebuah dokumen dengan label l . $\sum_{w' \in vw'ct}$ adalah jumlah total dari keseluruhan kata yang terdapat pada keseluruhan dokumen training, B' adalah jumlah idf secara keseluruhan. Jika menggunakan metode pembobotan tf idf, maka akan memberikan nilai 0 pada sebagian besar kata.

Untuk menghilangkan nol tersebut maka digunakan *laplace smoothing*, yaitu dengan menambahkan nilai 1 pada setiap W_{lt} dari perhitungan conditional probability. Maka persamaan untuk laplace smoothing sebagai berikut:

$$P(t|l) = \frac{W_{lt} + 1}{(\sum_{w' \in vw'ct} + B')} \quad (4)$$

Ketika nilai conditional probability sudah diketahui untuk setiap kata di masing-masing kelas selanjutnya dilakukan testing dengan data yang sama untuk menguji ketepatan klasifikasi dengan cara mengalikan *prior probability* beserta nilai *conditional probability* dari seluruh kata dalam kelas. Untuk kata yang muncul lebih dari satu kali, pangkatkan nilai *conditional probability* yang sudah digabungkan dengan *laplace smoothing*.

2.2.8. TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) adalah metode yang digunakan untuk menghitung bobotan setiap kata yang telah diekstrak. Penggunaan metode ini umumnya dilakukan untuk menghitung kata umum yang ada pada information retrieval. *TF-IDF* merupakan metode yang mengintegrasikan model *TF* dan *IDF* (Akbari, Novianty & Setianingsih, 2017). Adapun rumus pembobotan kata dengan *TF-IDF* adalah sebagai berikut.

$$tf - idf = tf_{t,d} * idf_t = tf(t, d) * \log \frac{N_d}{df_t} + 1 \quad (5)$$

Keterangan :

N_d = Jumlah semua dokumen

df_t = Jumlah term yang muncul dalam keseluruhan dokumen

$tf_{t,d}$ = Jumlah kemunculan term (t) dalam dokumen (d)

idf_t = Nilai sebaran term pada koleksi yang bersangkutan

2.2.9. *Split Validation*

Split validation merupakan sebuah operator yang menerapkan *simple validation* yakni memisahkan (*split*) data set secara acak kedalam sebuah data set training dan dataset testing dan evaluasi model yang terbentuk (Rapid Miner, 2014). Dengan menggunakan *Split Validation* akan dilakukan percobaan *training* berdasarkan *split ratio* yang telah ditentukan sebelumnya, untuk kemudian sisa dari *split ratio* data *training* akan dianggap sebagai data *testing*. Data *training* adalah data yang akan dipakai dalam melakukan pembelajaran sedangkan data *testing* adalah data yang belum pernah dipakai sebagai pembelajaran dan akan berfungsi sebagai data pengujian kebenaran atau keakurasian hasil pembelajaran

2.2.10. *Confusion Matrix*

Confusion matrix merupakan salah satu *tools* penting dalam metode visualisasi yang digunakan pada mesin pembelajaran yang biasanya memuat dua kelas atau lebih (Manning, dkk, 2009). Sebuah matriks dari prediksi yang akan dibandingkan dengan kelas yang asli dari inputan atau dengan kata lain berisi informasi nilai aktual dan prediksi pada klasifikasi. Tabel *Confusion Matrix* ditunjukkan dalam tabel 2.4.

Tabel 2.4 Tabel Confusion Matrix

	Kelas Prediksi 1 (Positif)	Kelas Prediksi 2 (Negatif)
Kelas Aktual 1 (positif)	TP (True Positif)	FN (False Negatif)
Kelas Aktual 2 (Negatif)	FP (False Positif)	TN (True Negatif)

Rumus untuk menghitung tingkat akurasi pada *confusion matrix* adalah :

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (6)$$

Keterangan :

TP = Prediksi positif benar karena kelas aktual positif

FP = Prediksi positif salah karena kelas aktual negatif

TN = Prediksi negatif benar karena kelas aktual negatif

FN = Prediksi negatif salah karena kelas aktual positif

2.2.11. Tukang Parkir

Juru parkir yang disebut juga sebagai tukang parkir adalah orang yang membantu mengatur kendaraan yang keluar masuk ke tempat parkir. Tukang parkir juga berfungsi untuk mengumpulkan biaya parkir dan memberikan karcis kepada pengguna parkir pada saat akan keluar dari ruang parkir. (Wikipedia, 2020)

BAB III

METODE PENELITIAN

3.1. Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah metode *Naïve Bayes* yaitu merupakan metode klasifikasi yang akan diuji keakuratannya dalam melakukan klasifikasi sentimen terhadap profesi Tukang Parkir di *platform* Twitter.

3.2. Alur Penelitian

Metode yang digunakan dalam penelitian ini adalah metode *Naïve Bayes* yaitu merupakan metode klasifikasi yang akan diuji keakuratannya dalam melakukan klasifikasi sentimen terhadap kata kunci Tukang Parkir di *platform* Twitter.

3.2.1. Studi Pustaka

Studi pustaka dilakukan untuk memahami dan memperdalam hal-hal yang berkaitan dengan penelitian, dengan cara mempelajari referensi berupa penelitian terdahulu, buku, serta literatur yang berkaitan dengan penelitian ini

3.2.2. Pengumpulan Data

Pengumpulan data dilakukan dengan mengumpulkan data Tweet berjumlah 5017 dalam rentang waktu mulai 1 November 2019 sampai 31 Desember 2019 yang akan digunakan sebagai data untuk membuat pemodelan, dan pengumpulan data tweet sejumlah 1000 tweet pada rentang waktu 7 -12 juli 2020 sebagai data implementasi. Adapun pengambilan data ini menggunakan *library twitterscrapper* dan twitter *API*.

3.2.3. Seleksi Data

Setelah data terkumpul, tahap awal penelitian dimulai dengan melakukan proses seleksi data. Pada tahap ini dilakukan proses menentukan data mana yang akan digunakan dalam proses penelitian, dan data mana yang tidak dipakai dalam proses penelitian. Pada tahap ini juga dilakukan proses pelabelan data yang dilakukan oleh 3 orang dengan ketentuan masing masing melabeli setiap tweet apakah tweet tersebut positif, negatif ,atau tidak relevan. Data yang duplikat dan tidak relevan nantinya tidak akan digunakan dalam penelitian ini.

3.2.4. *Preprocessing Data*

Dalam proses *preprocessing* data terdapat beberapa tahapan yang dilakukan diantaranya *cleansing* yaitu membuang tanda baca dan konversi menjadi huruf kecil (*lowercase*), lalu *slangword remover* yaitu perubahan kata tidak baku musiman menjadi kata baku, setelah itu dilakukan proses *stopword removal* yaitu proses pembersihan kata yang sering muncul dan bernilai kecil, lalu dilanjutkan dengan proses *stemming* yaitu perubahan kata menjadi bentuk kata dasar (*stem*),

3.2.5. Analisa

Setelah melalui tahap *preprocessing*, lalu dilakukan proses pembobotan kata (*term weighting*) dengan menggunakan algoritma *TF-IDF*. Kemudian data yang sudah melalui proses pembobotan akan diolah dengan *Naïve Bayes Classifier* sehingga menghasilkan model pembelajaran yang nantinya akan digunakan untuk implementasi pada data baru yang tidak diberikan label untuk diklasifikasi secara otomatis. Sebelum memasuki tahap implementasi, dalam tahap ini juga dilakukan validasi dengan teknik *split validation* untuk mengetahui kinerja model dengan

menghitung akurasi, serta dilakukan evaluasi model klasifikasi dengan metode *confusion matrix* untuk menghitung performa dari model klasifikasi yang sudah dibuat.

3.2.6. Implementasi

Berdasarkan data latih yang sudah diolah menjadi model pembelajaran menggunakan *Naïve Bayes Classifier*, selanjutnya akan dilakukan proses klasifikasi sentimen secara otomatis terhadap data baru yang belum diketahui sentimennya ke dalam kelompok sentimen negatif dan positif dengan menggunakan model yang sudah dibuat pada tahap analisis.

3.2.7. Pembuatan Laporan

Pembuatan laporan dilakukan bersamaan dengan dilakukannya penelitian, sehingga tidak dibutuh waktu berbeda dalam pengerjaannya.

3.3. Perangkat Penelitian

Alat dan Bahan yang diperlukan dalam penelitian ini berupa perangkat keras dan perangkat lunak. Perangkat yang digunakan dalam penelitian ini adalah:

3.3.1. Perangkat Keras

Perangkat keras yang dibutuhkan penulis untuk merancang dan membangun sistem perbandingan ini memiliki spesifikasi berikut ini:

1. Laptop Processor AMD A8
2. RAM 4GB

3.3.2. Perangkat Lunak

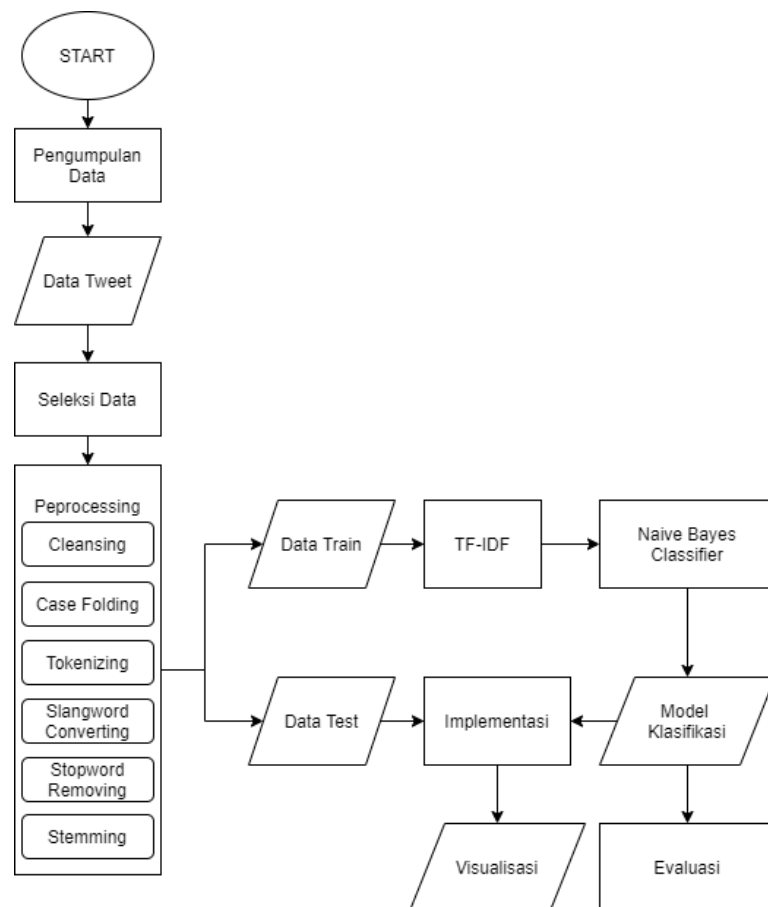
Perangkat lunak yang digunakan untuk penelitian ini adalah:

1. Sistem Operasi Windows 10 Professional x64
2. Bahasa pemrograman *python*
3. Anaconda3
4. Jupyter Notebook
5. Library twitterscrapper

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini, akan dianalisis secara rinci terkait masing-masing algoritma dan perbandingan keduanya berdasarkan Gambar 4.1:



Gambar 4.1 *Flowchart* Proses Bisnis

4.1. Pengumpulan Data

Pengumpulan data dilakukan dengan *crawling* tweet menggunakan *library* *twitterscrapper* dengan kata kunci “Tukang Parkir” yang diambil dalam rentang waktu 1 November 2019 sampai 31 Desember 2019, sejumlah 5017 data. Data

tersebut berisikan informasi tweet dengan berbagai macam atribut diantaranya id tweet, username, waktu tweet dibuat, isi tweet, dan masih banyak lagi. Dalam penelitian ini, hanya diambil data yang memiliki peran dalam proses klasifikasi. Untuk atribut yang diambil hanya username dan isi tweet, data disimpan dalam format file .csv.

Tabel 4.1 Contoh Data Crawling

username	tweet
Lip	Pacaran itu sama dengan tukang parkir. gak selamanya harus "Terus... Terus..." Pasti ada saat dia bilang "Yak stop!"
s123	Bisa jd inspirasi, jgn masih muda ud mlas kerja malah milih ngamen, dan jd tukang parkir https://twitter.com/KompasTV/status/1196380024871563264 !
Paseo	Salut ma tukang parkir alfamidi, cekatan dan ramah bgt. jadi keinget ma alm bapak
kuynom	Oalah anda tukang parkir di indomart saya ya minta yg suka minta duitnya maksa Owkwkwk
rakuyan	Cuma gara gara peluitnya tukang parkir aku bisa ngakak lho @luamsa_ Bahagia itu sederhana
Yauda Sih	Emg susah kok menghapus korupsi di negara ini. Orang dari tukang parkir aja ada yg doyan korupsi. Apalagi petinggi.

4.2. Pelabelan dan Seleksi Data

Data yang didapatkan dari Twitter masih berupa data mentah yang perlu dilakukan pemilahan dengan membuang data yang tidak sesuai seperti data duplikat dan tidak sesuai. Data yang sudah diseleksi duplikatnya lalu dilakukan proses pelabelan secara manual oleh 3 orang yang berbeda dengan mengelompokkan setiap tweet ke dalam tiga kelas yaitu positif dan negatif negatif dan tidak relevan.

Pemberian label negatif didasari oleh opini yang negatif atau ketidaksetujuan dalam konteks profesi tukang parkir, pemberian label positif didasari oleh opini yang positif atau kesetujuan dalam konteks profesi tukang parkir. Tweet yang

tidak menunjukkan opini positif atau negatif tidak digunakan dan diberi label tidak relevan. Data yang digunakan dalam penelitian ini hanya menggunakan data berlabel positif dan negatif.

Tabel 4.2 Contoh Data dengan Label

tweet	label
Bisa jd inspirasi, jgn masih muda ud mlas kerja malah milih ngamen, dan jd tukang parkir https://twitter.com/KompasTV/status/1196380024871563264 , ¹ ² ³ ⁴ ⁵ ⁶ ⁷ ⁸ ⁹ ¹⁰ ¹¹ ¹² ¹³ ¹⁴ ¹⁵ ¹⁶ ¹⁷ ¹⁸ ¹⁹ ²⁰ ²¹ ²² ²³ ²⁴ ²⁵ ²⁶ ²⁷ ²⁸ ²⁹ ³⁰ ³¹ ³² ³³ ³⁴ ³⁵ ³⁶ ³⁷ ³⁸ ³⁹ ⁴⁰ ⁴¹ ⁴² ⁴³ ⁴⁴ ⁴⁵ ⁴⁶ ⁴⁷ ⁴⁸ ⁴⁹ ⁵⁰ ⁵¹ ⁵² ⁵³ ⁵⁴ ⁵⁵ ⁵⁶ ⁵⁷ ⁵⁸ ⁵⁹ ⁶⁰ ⁶¹ ⁶² ⁶³ ⁶⁴ ⁶⁵ ⁶⁶ ⁶⁷ ⁶⁸ ⁶⁹ ⁷⁰ ⁷¹ ⁷² ⁷³ ⁷⁴ ⁷⁵ ⁷⁶ ⁷⁷ ⁷⁸ ⁷⁹ ⁸⁰ ⁸¹ ⁸² ⁸³ ⁸⁴ ⁸⁵ ⁸⁶ ⁸⁷ ⁸⁸ ⁸⁹ ⁹⁰ ⁹¹ ⁹² ⁹³ ⁹⁴ ⁹⁵ ⁹⁶ ⁹⁷ ⁹⁸ ⁹⁹ ¹⁰⁰ ¹⁰¹ ¹⁰² ¹⁰³ ¹⁰⁴ ¹⁰⁵ ¹⁰⁶ ¹⁰⁷ ¹⁰⁸ ¹⁰⁹ ¹¹⁰ ¹¹¹ ¹¹² ¹¹³ ¹¹⁴ ¹¹⁵ ¹¹⁶ ¹¹⁷ ¹¹⁸ ¹¹⁹ ¹²⁰ ¹²¹ ¹²² ¹²³ ¹²⁴ ¹²⁵ ¹²⁶ ¹²⁷ ¹²⁸ ¹²⁹ ¹³⁰ ¹³¹ ¹³² ¹³³ ¹³⁴ ¹³⁵ ¹³⁶ ¹³⁷ ¹³⁸ ¹³⁹ ¹⁴⁰ ¹⁴¹ ¹⁴² ¹⁴³ ¹⁴⁴ ¹⁴⁵ ¹⁴⁶ ¹⁴⁷ ¹⁴⁸ ¹⁴⁹ ¹⁵⁰ ¹⁵¹ ¹⁵² ¹⁵³ ¹⁵⁴ ¹⁵⁵ ¹⁵⁶ ¹⁵⁷ ¹⁵⁸ ¹⁵⁹ ¹⁶⁰ ¹⁶¹ ¹⁶² ¹⁶³ ¹⁶⁴ ¹⁶⁵ ¹⁶⁶ ¹⁶⁷ ¹⁶⁸ ¹⁶⁹ ¹⁷⁰ ¹⁷¹ ¹⁷² ¹⁷³ ¹⁷⁴ ¹⁷⁵ ¹⁷⁶ ¹⁷⁷ ¹⁷⁸ ¹⁷⁹ ¹⁸⁰ ¹⁸¹ ¹⁸² ¹⁸³ ¹⁸⁴ ¹⁸⁵ ¹⁸⁶ ¹⁸⁷ ¹⁸⁸ ¹⁸⁹ ¹⁹⁰ ¹⁹¹ ¹⁹² ¹⁹³ ¹⁹⁴ ¹⁹⁵ ¹⁹⁶ ¹⁹⁷ ¹⁹⁸ ¹⁹⁹ ²⁰⁰ ²⁰¹ ²⁰² ²⁰³ ²⁰⁴ ²⁰⁵ ²⁰⁶ ²⁰⁷ ²⁰⁸ ²⁰⁹ ²¹⁰ ²¹¹ ²¹² ²¹³ ²¹⁴ ²¹⁵ ²¹⁶ ²¹⁷ ²¹⁸ ²¹⁹ ²²⁰ ²²¹ ²²² ²²³ ²²⁴ ²²⁵ ²²⁶ ²²⁷ ²²⁸ ²²⁹ ²³⁰ ²³¹ ²³² ²³³ ²³⁴ ²³⁵ ²³⁶ ²³⁷ ²³⁸ ²³⁹ ²⁴⁰ ²⁴¹ ²⁴² ²⁴³ ²⁴⁴ ²⁴⁵ ²⁴⁶ ²⁴⁷ ²⁴⁸ ²⁴⁹ ²⁵⁰ ²⁵¹ ²⁵² ²⁵³ ²⁵⁴ ²⁵⁵ ²⁵⁶ ²⁵⁷ ²⁵⁸ ²⁵⁹ ²⁶⁰ ²⁶¹ ²⁶² ²⁶³ ²⁶⁴ ²⁶⁵ ²⁶⁶ ²⁶⁷ ²⁶⁸ ²⁶⁹ ²⁷⁰ ²⁷¹ ²⁷² ²⁷³ ²⁷⁴ ²⁷⁵ ²⁷⁶ ²⁷⁷ ²⁷⁸ ²⁷⁹ ²⁸⁰ ²⁸¹ ²⁸² ²⁸³ ²⁸⁴ ²⁸⁵ ²⁸⁶ ²⁸⁷ ²⁸⁸ ²⁸⁹ ²⁹⁰ ²⁹¹ ²⁹² ²⁹³ ²⁹⁴ ²⁹⁵ ²⁹⁶ ²⁹⁷ ²⁹⁸ ²⁹⁹ ³⁰⁰ ³⁰¹ ³⁰² ³⁰³ ³⁰⁴ ³⁰⁵ ³⁰⁶ ³⁰⁷ ³⁰⁸ ³⁰⁹ ³¹⁰ ³¹¹ ³¹² ³¹³ ³¹⁴ ³¹⁵ ³¹⁶ ³¹⁷ ³¹⁸ ³¹⁹ ³²⁰ ³²¹ ³²² ³²³ ³²⁴ ³²⁵ ³²⁶ ³²⁷ ³²⁸ ³²⁹ ³³⁰ ³³¹ ³³² ³³³ ³³⁴ ³³⁵ ³³⁶ ³³⁷ ³³⁸ ³³⁹ ³⁴⁰ ³⁴¹ ³⁴² ³⁴³ ³⁴⁴ ³⁴⁵ ³⁴⁶ ³⁴⁷ ³⁴⁸ ³⁴⁹ ³⁵⁰ ³⁵¹ ³⁵² ³⁵³ ³⁵⁴ ³⁵⁵ ³⁵⁶ ³⁵⁷ ³⁵⁸ ³⁵⁹ ³⁶⁰ ³⁶¹ ³⁶² ³⁶³ ³⁶⁴ ³⁶⁵ ³⁶⁶ ³⁶⁷ ³⁶⁸ ³⁶⁹ ³⁷⁰ ³⁷¹ ³⁷² ³⁷³ ³⁷⁴ ³⁷⁵ ³⁷⁶ ³⁷⁷ ³⁷⁸ ³⁷⁹ ³⁸⁰ ³⁸¹ ³⁸² ³⁸³ ³⁸⁴ ³⁸⁵ ³⁸⁶ ³⁸⁷ ³⁸⁸ ³⁸⁹ ³⁹⁰ ³⁹¹ ³⁹² ³⁹³ ³⁹⁴ ³⁹⁵ ³⁹⁶ ³⁹⁷ ³⁹⁸ ³⁹⁹ ⁴⁰⁰ ⁴⁰¹ ⁴⁰² ⁴⁰³ ⁴⁰⁴ ⁴⁰⁵ ⁴⁰⁶ ⁴⁰⁷ ⁴⁰⁸ ⁴⁰⁹ ⁴¹⁰ ⁴¹¹ ⁴¹² ⁴¹³ ⁴¹⁴ ⁴¹⁵ ⁴¹⁶ ⁴¹⁷ ⁴¹⁸ ⁴¹⁹ ⁴²⁰ ⁴²¹ ⁴²² ⁴²³ ⁴²⁴ ⁴²⁵ ⁴²⁶ ⁴²⁷ ⁴²⁸ ⁴²⁹ ⁴³⁰ ⁴³¹ ⁴³² ⁴³³ ⁴³⁴ ⁴³⁵ ⁴³⁶ ⁴³⁷ ⁴³⁸ ⁴³⁹ ⁴⁴⁰ ⁴⁴¹ ⁴⁴² ⁴⁴³ ⁴⁴⁴ ⁴⁴⁵ ⁴⁴⁶ ⁴⁴⁷ ⁴⁴⁸ ⁴⁴⁹ ⁴⁵⁰ ⁴⁵¹ ⁴⁵² ⁴⁵³ ⁴⁵⁴ ⁴⁵⁵ ⁴⁵⁶ ⁴⁵⁷ ⁴⁵⁸ ⁴⁵⁹ ⁴⁶⁰ ⁴⁶¹ ⁴⁶² ⁴⁶³ ⁴⁶⁴ ⁴⁶⁵ ⁴⁶⁶ ⁴⁶⁷ ⁴⁶⁸ ⁴⁶⁹ ⁴⁷⁰ ⁴⁷¹ ⁴⁷² ⁴⁷³ ⁴⁷⁴ ⁴⁷⁵ ⁴⁷⁶ ⁴⁷⁷ ⁴⁷⁸ ⁴⁷⁹ ⁴⁸⁰ ⁴⁸¹ ⁴⁸² ⁴⁸³ ⁴⁸⁴ ⁴⁸⁵ ⁴⁸⁶ ⁴⁸⁷ ⁴⁸⁸ ⁴⁸⁹ ⁴⁹⁰ ⁴⁹¹ ⁴⁹² ⁴⁹³ ⁴⁹⁴ ⁴⁹⁵ ⁴⁹⁶ ⁴⁹⁷ ⁴⁹⁸ ⁴⁹⁹ ⁵⁰⁰ ⁵⁰¹ ⁵⁰² ⁵⁰³ ⁵⁰⁴ ⁵⁰⁵ ⁵⁰⁶ ⁵⁰⁷ ⁵⁰⁸ ⁵⁰⁹ ⁵¹⁰ ⁵¹¹ ⁵¹² ⁵¹³ ⁵¹⁴ ⁵¹⁵ ⁵¹⁶ ⁵¹⁷ ⁵¹⁸ ⁵¹⁹ ⁵²⁰ ⁵²¹ ⁵²² ⁵²³ ⁵²⁴ ⁵²⁵ ⁵²⁶ ⁵²⁷ ⁵²⁸ ⁵²⁹ ⁵³⁰ ⁵³¹ ⁵³² ⁵³³ ⁵³⁴ ⁵³⁵ ⁵³⁶ ⁵³⁷ ⁵³⁸ ⁵³⁹ ⁵⁴⁰ ⁵⁴¹ ⁵⁴² ⁵⁴³ ⁵⁴⁴ ⁵⁴⁵ ⁵⁴⁶ ⁵⁴⁷ ⁵⁴⁸ ⁵⁴⁹ ⁵⁵⁰ ⁵⁵¹ ⁵⁵² ⁵⁵³ ⁵⁵⁴ ⁵⁵⁵ ⁵⁵⁶ ⁵⁵⁷ ⁵⁵⁸ ⁵⁵⁹ ⁵⁶⁰ ⁵⁶¹ ⁵⁶² ⁵⁶³ ⁵⁶⁴ ⁵⁶⁵ ⁵⁶⁶ ⁵⁶⁷ ⁵⁶⁸ ⁵⁶⁹ ⁵⁷⁰ ⁵⁷¹ ⁵⁷² ⁵⁷³ ⁵⁷⁴ ⁵⁷⁵ ⁵⁷⁶ ⁵⁷⁷ ⁵⁷⁸ ⁵⁷⁹ ⁵⁸⁰ ⁵⁸¹ ⁵⁸² ⁵⁸³ ⁵⁸⁴ ⁵⁸⁵ ⁵⁸⁶ ⁵⁸⁷ ⁵⁸⁸ ⁵⁸⁹ ⁵⁹⁰ ⁵⁹¹ ⁵⁹² ⁵⁹³ ⁵⁹⁴ ⁵⁹⁵ ⁵⁹⁶ ⁵⁹⁷ ⁵⁹⁸ ⁵⁹⁹ ⁶⁰⁰ ⁶⁰¹ ⁶⁰² ⁶⁰³ ⁶⁰⁴ ⁶⁰⁵ ⁶⁰⁶ ⁶⁰⁷ ⁶⁰⁸ ⁶⁰⁹ ⁶¹⁰ ⁶¹¹ ⁶¹² ⁶¹³ ⁶¹⁴ ⁶¹⁵ ⁶¹⁶ ⁶¹⁷ ⁶¹⁸ ⁶¹⁹ ⁶²⁰ ⁶²¹ ⁶²² ⁶²³ ⁶²⁴ ⁶²⁵ ⁶²⁶ ⁶²⁷ ⁶²⁸ ⁶²⁹ ⁶³⁰ ⁶³¹ ⁶³² ⁶³³ ⁶³⁴ ⁶³⁵ ⁶³⁶ ⁶³⁷ ⁶³⁸ ⁶³⁹ ⁶⁴⁰ ⁶⁴¹ ⁶⁴² ⁶⁴³ ⁶⁴⁴ ⁶⁴⁵ ⁶⁴⁶ ⁶⁴⁷ ⁶⁴⁸ ⁶⁴⁹ ⁶⁵⁰ ⁶⁵¹ ⁶⁵² ⁶⁵³ ⁶⁵⁴ ⁶⁵⁵ ⁶⁵⁶ ⁶⁵⁷ ⁶⁵⁸ ⁶⁵⁹ ⁶⁶⁰ ⁶⁶¹ ⁶⁶² ⁶⁶³ ⁶⁶⁴ ⁶⁶⁵ ⁶⁶⁶ ⁶⁶⁷ ⁶⁶⁸ ⁶⁶⁹ ⁶⁷⁰ ⁶⁷¹ ⁶⁷² ⁶⁷³ ⁶⁷⁴ ⁶⁷⁵ ⁶⁷⁶ ⁶⁷⁷ ⁶⁷⁸ ⁶⁷⁹ ⁶⁸⁰ ⁶⁸¹ ⁶⁸² ⁶⁸³ ⁶⁸⁴ ⁶⁸⁵ ⁶⁸⁶ ⁶⁸⁷ ⁶⁸⁸ ⁶⁸⁹ ⁶⁹⁰ ⁶⁹¹ ⁶⁹² ⁶⁹³ ⁶⁹⁴ ⁶⁹⁵ ⁶⁹⁶ ⁶⁹⁷ ⁶⁹⁸ ⁶⁹⁹ ⁷⁰⁰ ⁷⁰¹ ⁷⁰² ⁷⁰³ ⁷⁰⁴ ⁷⁰⁵ ⁷⁰⁶ ⁷⁰⁷ ⁷⁰⁸ ⁷⁰⁹ ⁷¹⁰ ⁷¹¹ ⁷¹² ⁷¹³ ⁷¹⁴ ⁷¹⁵ ⁷¹⁶ ⁷¹⁷ ⁷¹⁸ ⁷¹⁹ ⁷²⁰ ⁷²¹ ⁷²² ⁷²³ ⁷²⁴ ⁷²⁵ ⁷²⁶ ⁷²⁷ ⁷²⁸ ⁷²⁹ ⁷³⁰ ⁷³¹ ⁷³² ⁷³³ ⁷³⁴ ⁷³⁵ ⁷³⁶ ⁷³⁷ ⁷³⁸ ⁷³⁹ ⁷⁴⁰ ⁷⁴¹ ⁷⁴² ⁷⁴³ ⁷⁴⁴ ⁷⁴⁵ ⁷⁴⁶ ⁷⁴⁷ ⁷⁴⁸ ⁷⁴⁹ ⁷⁵⁰ ⁷⁵¹ ⁷⁵² ⁷⁵³ ⁷⁵⁴ ⁷⁵⁵ ⁷⁵⁶ ⁷⁵⁷ ⁷⁵⁸ ⁷⁵⁹ ⁷⁶⁰ ⁷⁶¹ ⁷⁶² ⁷⁶³ ⁷⁶⁴ ⁷⁶⁵ ⁷⁶⁶ ⁷⁶⁷ ⁷⁶⁸ ⁷⁶⁹ ⁷⁷⁰ ⁷⁷¹ ⁷⁷² ⁷⁷³ ⁷⁷⁴ ⁷⁷⁵ ⁷⁷⁶ ⁷⁷⁷ ⁷⁷⁸ ⁷⁷⁹ ⁷⁸⁰ ⁷⁸¹ ⁷⁸² ⁷⁸³ ⁷⁸⁴ ⁷⁸⁵ ⁷⁸⁶ ⁷⁸⁷ ⁷⁸⁸ ⁷⁸⁹ ⁷⁹⁰ ⁷⁹¹ ⁷⁹² ⁷⁹³ ⁷⁹⁴ ⁷⁹⁵ ⁷⁹⁶ ⁷⁹⁷ ⁷⁹⁸ ⁷⁹⁹ ⁸⁰⁰ ⁸⁰¹ ⁸⁰² ⁸⁰³ ⁸⁰⁴ ⁸⁰⁵ ⁸⁰⁶ ⁸⁰⁷ ⁸⁰⁸ ⁸⁰⁹ ⁸¹⁰ ⁸¹¹ ⁸¹² ⁸¹³ ⁸¹⁴ ⁸¹⁵ ⁸¹⁶ ⁸¹⁷ ⁸¹⁸ ⁸¹⁹ ⁸²⁰ ⁸²¹ ⁸²² ⁸²³ ⁸²⁴ ⁸²⁵ ⁸²⁶ ⁸²⁷ ⁸²⁸ ⁸²⁹ ⁸³⁰ ⁸³¹ ⁸³² ⁸³³ ⁸³⁴ ⁸³⁵ ⁸³⁶ ⁸³⁷ ⁸³⁸ ⁸³⁹ ⁸⁴⁰ ⁸⁴¹ ⁸⁴² ⁸⁴³ ⁸⁴⁴ ⁸⁴⁵ ⁸⁴⁶ ⁸⁴⁷ ⁸⁴⁸ ⁸⁴⁹ ⁸⁵⁰ ⁸⁵¹ ⁸⁵² ⁸⁵³ ⁸⁵⁴ ⁸⁵⁵ ⁸⁵⁶ ⁸⁵⁷ ⁸⁵⁸ ⁸⁵⁹ ⁸⁶⁰ ⁸⁶¹ ⁸⁶² ⁸⁶³ ⁸⁶⁴ ⁸⁶⁵ ⁸⁶⁶ ⁸⁶⁷ ⁸⁶⁸ ⁸⁶⁹ ⁸⁷⁰ ⁸⁷¹ ⁸⁷² ⁸⁷³ ⁸⁷⁴ ⁸⁷⁵ ⁸⁷⁶ ⁸⁷⁷ ⁸⁷⁸ ⁸⁷⁹ ⁸⁸⁰ ⁸⁸¹ ⁸⁸² ⁸⁸³ ⁸⁸⁴ ⁸⁸⁵ ⁸⁸⁶ ⁸⁸⁷ ⁸⁸⁸ ⁸⁸⁹ ⁸⁹⁰ ⁸⁹¹ ⁸⁹² ⁸⁹³ ⁸⁹⁴ ⁸⁹⁵ ⁸⁹⁶ ⁸⁹⁷ ⁸⁹⁸ ⁸⁹⁹ ⁹⁰⁰ ⁹⁰¹ ⁹⁰² ⁹⁰³ ⁹⁰⁴ ⁹⁰⁵ ⁹⁰⁶ ⁹⁰⁷ ⁹⁰⁸ ⁹⁰⁹ ⁹¹⁰ ⁹¹¹ ⁹¹² ⁹¹³ ⁹¹⁴ ⁹¹⁵ ⁹¹⁶ ⁹¹⁷ ⁹¹⁸ ⁹¹⁹ ⁹²⁰ ⁹²¹ ⁹²² ⁹²³ ⁹²⁴ ⁹²⁵ ⁹²⁶ ⁹²⁷ ⁹²⁸ ⁹²⁹ ⁹³⁰ ⁹³¹ ⁹³² ⁹³³ ⁹³⁴ ⁹³⁵ ⁹³⁶ ⁹³⁷ ⁹³⁸ ⁹³⁹ ⁹⁴⁰ ⁹⁴¹ ⁹⁴² ⁹⁴³ ⁹⁴⁴ ⁹⁴⁵ ⁹⁴⁶ ⁹⁴⁷ ⁹⁴⁸ ⁹⁴⁹ ⁹⁵⁰ ⁹⁵¹ ⁹⁵² ⁹⁵³ ⁹⁵⁴ ⁹⁵⁵ ⁹⁵⁶ ⁹⁵⁷ ⁹⁵⁸ ⁹⁵⁹ ⁹⁶⁰ ⁹⁶¹ ⁹⁶² ⁹⁶³ ⁹⁶⁴ ⁹⁶⁵ ⁹⁶⁶ ⁹⁶⁷ ⁹⁶⁸ ⁹⁶⁹ ⁹⁷⁰ ⁹⁷¹ ⁹⁷² ⁹⁷³ ⁹⁷⁴ ⁹⁷⁵ ⁹⁷⁶ ⁹⁷⁷ ⁹⁷⁸ ⁹⁷⁹ ⁹⁸⁰ ⁹⁸¹ ⁹⁸² ⁹⁸³ ⁹⁸⁴ ⁹⁸⁵ ⁹⁸⁶ ⁹⁸⁷ ⁹⁸⁸ ⁹⁸⁹ ⁹⁹⁰ ⁹⁹¹ ⁹⁹² ⁹⁹³ ⁹⁹⁴ ⁹⁹⁵ ⁹⁹⁶ ⁹⁹⁷ ⁹⁹⁸ ⁹⁹⁹ ¹⁰⁰⁰ ¹⁰⁰¹ ¹⁰⁰² ¹⁰⁰³ ¹⁰⁰⁴ ¹⁰⁰⁵ ¹⁰⁰⁶ ¹⁰⁰⁷ ¹⁰⁰⁸ ¹⁰⁰⁹ ¹⁰¹⁰ ¹⁰¹¹ ¹⁰¹² ¹⁰¹³ ¹⁰¹⁴ ¹⁰¹⁵ ¹⁰¹⁶ ¹⁰¹⁷ ¹⁰¹⁸ ¹⁰¹⁹ ¹⁰²⁰ ¹⁰²¹ ¹⁰²² ¹⁰²³ ¹⁰²⁴ ¹⁰²⁵ ¹⁰²⁶ ¹⁰²⁷ ¹⁰²⁸ ¹⁰²⁹ ¹⁰³⁰ ¹⁰³¹ ¹⁰³² ¹⁰³³ ¹⁰³⁴ ¹⁰³⁵ ¹⁰³⁶ ¹⁰³⁷ ¹⁰³⁸ ¹⁰³⁹ ¹⁰⁴⁰ ¹⁰⁴¹ ¹⁰⁴² ¹⁰⁴³ ¹⁰⁴⁴ ¹⁰⁴⁵ ¹⁰⁴⁶ ¹⁰⁴⁷ ¹⁰⁴⁸ ¹⁰⁴⁹ ¹⁰⁵⁰ ¹⁰⁵¹ ¹⁰⁵² ¹⁰⁵³ ¹⁰⁵⁴ ¹⁰⁵⁵ ¹⁰⁵⁶ ¹⁰⁵⁷ ¹⁰⁵⁸ ¹⁰⁵⁹ ¹⁰⁶⁰ ¹⁰⁶¹ ¹⁰⁶² ¹⁰⁶³ ¹⁰⁶⁴ ¹⁰⁶⁵ ¹⁰⁶⁶ ¹⁰⁶⁷ ¹⁰⁶⁸ ¹⁰⁶⁹ ¹⁰⁷⁰ ¹⁰⁷¹ ¹⁰⁷² ¹⁰⁷³ ¹⁰⁷⁴ ¹⁰⁷⁵ ¹⁰⁷⁶ ¹⁰⁷⁷ ¹⁰⁷⁸ ¹⁰⁷⁹ ¹⁰⁸⁰ ¹⁰⁸¹ ¹⁰⁸² ¹⁰⁸³ ¹⁰⁸⁴ ¹⁰⁸⁵ ¹⁰⁸⁶ ¹⁰⁸⁷ ¹⁰⁸⁸ ¹⁰⁸⁹ ¹⁰⁹⁰ ¹⁰⁹¹ ¹⁰⁹² ¹⁰⁹³ ¹⁰⁹⁴ ¹⁰⁹⁵ ¹⁰⁹⁶ ¹⁰⁹⁷ ¹⁰⁹⁸ ¹⁰⁹⁹ ¹¹⁰⁰ ¹¹⁰¹ ¹¹⁰² ¹¹⁰³ ¹¹⁰⁴ ¹¹⁰⁵ ¹¹⁰⁶ ¹¹⁰⁷ ¹¹⁰⁸ ¹¹⁰⁹ ¹¹¹⁰ ¹¹¹¹ ¹¹¹² ¹¹¹³ ¹¹¹⁴ ¹¹¹⁵ ¹¹¹⁶ ¹¹¹⁷ ¹¹¹⁸ ¹¹¹⁹ ¹¹²⁰ ¹¹²¹ ¹¹²² ¹¹²³ ¹¹²⁴ ¹¹²⁵ ¹¹²⁶ ¹¹²⁷ ¹¹²⁸ ¹¹²⁹ ¹¹³⁰ ¹¹³¹ ¹¹³² ¹¹³³ ¹¹³⁴ ¹¹³⁵ ¹¹³⁶ ¹¹³⁷ ¹¹³⁸ ¹¹³⁹ ¹¹⁴⁰ ¹¹⁴¹ ¹¹⁴² ¹¹⁴³ ¹¹⁴⁴ ¹¹⁴⁵ ¹¹⁴⁶ ¹¹⁴⁷ ¹¹⁴⁸ ¹¹⁴⁹ ¹¹⁵⁰ ¹¹⁵¹ ¹¹⁵² ¹¹⁵³ ¹¹⁵⁴ ¹¹⁵⁵ ¹¹⁵⁶ ¹¹⁵⁷ ¹¹⁵⁸ ¹¹⁵⁹ ¹¹⁶⁰ ¹¹⁶¹ ¹¹⁶² ¹¹⁶³ ¹¹⁶⁴ ¹¹⁶⁵ ¹¹⁶⁶ ¹¹⁶⁷ ¹¹⁶⁸ ¹¹⁶⁹ ¹¹⁷⁰ ¹¹⁷¹ ¹¹⁷² ¹¹⁷³ ¹¹⁷⁴ ¹¹⁷⁵ ¹¹⁷⁶ ¹¹⁷⁷ ¹¹⁷⁸ ¹¹⁷⁹ ¹¹⁸⁰ ¹¹⁸¹ ¹¹⁸² ¹¹⁸³ ¹¹⁸⁴ ¹¹⁸⁵ ¹¹⁸⁶ ¹¹⁸⁷ ¹¹⁸⁸ ¹¹⁸⁹ ¹¹⁹⁰ ¹¹⁹¹ ¹¹⁹² ¹¹⁹³ ¹¹⁹⁴ ¹¹⁹⁵ ¹¹⁹⁶ ¹¹⁹⁷ ¹¹⁹⁸ ¹¹⁹⁹ ¹²⁰⁰ ¹²⁰¹ ¹²⁰² ¹²⁰³ ¹²⁰⁴ ¹²⁰⁵ ¹²⁰⁶ ¹²⁰⁷ ¹²⁰⁸ ¹²⁰⁹ ¹²¹⁰ ¹²¹¹ ¹²¹² ¹²¹³ ¹²¹⁴ ¹²¹⁵ ¹²¹⁶ ¹²¹⁷ ¹²¹⁸ ¹²¹⁹ ¹²²⁰ ¹²²¹ ¹²²² ¹²²³ ¹²²⁴ ¹²²⁵ ¹²²⁶ ¹²²⁷ ¹²²⁸ ¹²²⁹ ¹²³⁰ ¹²³¹ ¹²³² ¹²³³ ¹²³⁴ ¹²³⁵ ¹²³⁶ ¹²³⁷ ¹²³⁸ ¹²³⁹ ¹²⁴⁰ ¹²⁴¹ ¹²⁴² ¹²⁴³ ¹²⁴⁴ ¹²⁴⁵ ¹²⁴⁶ ¹²⁴⁷ ¹²⁴⁸ ¹²⁴⁹ ¹²⁵⁰ ¹²⁵¹ ¹²⁵² ¹²⁵³ ¹²⁵⁴ ¹²⁵⁵ ¹²⁵⁶ ¹²⁵⁷ ¹²⁵⁸ ¹²⁵⁹ ¹²⁶⁰ ¹²⁶¹ ¹²⁶² ¹²⁶³ ¹²⁶⁴ ¹²⁶⁵ ¹²⁶⁶ ¹²⁶⁷ ¹²⁶⁸ ¹²⁶⁹ ¹²⁷⁰ ¹²⁷¹ ¹²⁷² ¹²⁷³ ¹²⁷⁴ ¹²⁷⁵ ¹²⁷⁶ ¹²⁷⁷ ¹²⁷⁸ ¹²⁷⁹ ¹²⁸⁰ ¹²⁸¹ ¹²⁸² ¹²⁸³ ¹²⁸⁴ ¹²⁸⁵ ¹²⁸⁶ ¹²⁸⁷ ¹²⁸⁸ ¹²⁸⁹ ¹²⁹⁰ ¹²⁹¹ ¹²⁹² ¹²⁹³ ¹²⁹⁴ ¹²⁹⁵ ¹²⁹⁶ ¹²⁹⁷ ¹²⁹⁸ ¹²⁹⁹ ¹³⁰⁰ ¹³⁰¹ ¹³⁰² ¹³⁰³ ¹³⁰⁴ ¹³⁰⁵ ¹³⁰⁶ ¹³⁰⁷ ¹³⁰⁸ ¹³⁰⁹ ¹³¹⁰ ¹³¹¹ ¹³¹² ¹³¹³ ¹³¹⁴ ¹³¹⁵ ¹³¹⁶ ¹³¹⁷ ¹³¹⁸ ¹³¹⁹ ¹³²⁰	

lainnya. Tahap *cleansing* dilakukan agar data yang masuk dalam proses pemodelan dapat diolah dengan baik. Pada tabel 4.3 disajikan perbandingan data sebelum dan sesudah dilakukan proses cleansing.

Tabel 4.3 Contoh Data Cleansing

sebelum	sesudah
Bisa jd inspirasi, jgn masih muda ud mlas kerja malah milih ngamen, dan jd tukang parkir https://twitter.com/KompasTV/status/1196380024871563264 , !	Bisa jd inspirasi jgn masih muda ud mlas kerja malah milih ngamen dan jd tukang parkir
Salut ma tukang parkir alfamidi, cekatan dan ramah bgt. jadi keinget ma alm bapak	Salut ma tukang parkir alfamidi cekatan dan ramah bgt jadi keinget ma alm bapak
Oalah anda tukang parkir di indomart saya ya minta yg suka minta duitnya maksa Owkwkwk	Oalah anda tukang parkir di indomart saya ya minta yg suka minta duitnya maksa Owkwkwk
Cuma gara gara peluitnya tukang parkir aku bisa ngakak lho @luamsa_ Bahagia itu sederhana	Cuma gara gara peluitnya tukang parkir aku bisa ngakak lho Bahagia itu sederhana
Emg susah kok menghapus korupsi di negara ini. Orang dari tukang parkir aja ada yg doyan korupsi. Apalagi petinggi.	Emg susah kok menghapus korupsi di negara ini Orang dari tukang parkir aja ada yg doyan korupsi. Apalagi petinggi

4.3.2. Case Folding

Tahap *Case Folding* bekerja dengan cara mengubah penggunaan huruf kapital dalam keseluruhan teks menjadi huruf kecil (*lowercase*). Tahapan ini perlu dilakukan karena dalam proses pembobotan teks penggunaan huruf kapital mempengaruhi perhitungan dalam term weighting menggunakan *TF-IDF*. Maka dari itu tujuan utama dilakukan proses *Case Folding* agar pada tahap *term weighting* bisa berjalan dengan baik.

Pada tabel 4.4 disajikan perbandingan data sebelum dan sesudah dilakukannya proses *Case Folding*.

Tabel 4.4 Contoh Data dengan *Casefolding*

sebelum	sesudah
Bisa jd inspirasi jgn masih muda ud mlas kerja malah milih ngamen dan jd tukang parkir	bisa jd inspirasi jgn masih muda ud mlas kerja malah milih ngamen dan jd tukang parkir
Salut ma tukang parkir alfamidi cekatan dan ramah bgt jadi keinget ma alm bapak	salut ma tukang parkir alfamidi cekatan dan ramah bgt jadi keinget ma alm bapak
Oalah anda tukang parkir di indomart saya ya minta yg suka minta duitnya maksa Owkwkwk	oalah anda tukang parkir di indomart saya ya minta yg suka minta duitnya maksa owkwkwk
Cuma gara gara peluitnya tukang parkir aku bisa ngakak lho Bahagia itu sederhana	cuma gara gara peluitnya tukang parkir aku bisa ngakak lho bahagia itu sederhana
Emg susah kok menghapus korupsi di negara ini Orang dari tukang parkir aja ada yg doyan korupsi. Apalagi petinggi	emg susah kok menghapus korupsi di negara ini orang dari tukang parkir aja ada yg doyan korupsi. apalagi petinggi

4.3.3. *Slangword Removal*

Dalam data tweet yang sudah terkumpul terdapat banyak sekali istilah atau kata yang tidak sesuai dengan kaidah (*slangword*) dalam bahasa Indonesia tapi masih cukup dapat dibaca dan dimengerti oleh manusia, namun untuk membuat pemodelan yang baik perlu dilakukan perubahan kata *slangword* menjadi kata yang sesuai dengan kaidah bahasa. Untuk merubah kata tidak baku menjadi baku, yang dilakukan pertama kali adalah memasukkan kamus kata-kata tidak baku (*slang word*) ke dalam system (Apriandi, dkk, 2016). Proses konversi *slangword* menggunakan kamus sumber terbuka yang bisa diunduh di internet.

Pada tabel 4.5 disajikan perbandingan data sebelum dan sesudah dilakukannya proses *Slangword Removal*.

Tabel 4.5 Contoh Data dengan *Slangword Removal*

sebelum	sesudah
bisa jd inspirasi jgn masih muda ud mlas kerja malah milih ngamen dan jd tukang parkir	bisa jadi inspirasi jangan masih muda udah malas kerja malah milih ngamen dan jadi tukang parkir
salut ma tukang parkir alfamidi cekatan dan ramah bgt jadi keinget ma alm bapak	salut sama tukang parkir alfamidi cekatan dan ramah banget jadi keinget sama almarhum bapak
oalah anda tukang parkir di indomart saya ya minta yg suka minta duitnya maksa owkwkwk	oalah anda tukang parkir di indomart saya ya minta yang suka minta duitnya maksa owkwkwk
cuma gara gara peluitnya tukang parkir aku bisa ngakak lho bahagia itu sederhana	cuma gara gara peluitnya tukang parkir aku bisa ngakak lho bahagia itu sederhana
emg susah kok menghapus korupsi di negara ini orang dari tukang parkir aja ada yg doyan korupsi. apalagi petinggi	emang susah kok menghapus korupsi di negara ini orang dari tukang parkir aja ada yang doyan korupsi. apalagi petinggi

4.3.4. *Stopword Removal*

Proses *stopword removal* dilakukan dengan cara membuang kata yang sering muncul dan mempunyai nilai yang kecil. *Stopword Removal* adalah proses untuk menghilangkan kata yang sering muncul namun tidak memiliki makna spesifik atau perannya tidak diperlukan dalam penelitian (Apriandi, dkk, 2016). Dalam pemrosesan *Stopword Removal*, penulis menggunakan *library* dari *Sastrawi*. Penghapusan *stopword* dalam *preprocessing* mampu meningkatkan performa klasifikasi yang dibuktikan dengan adanya peningkatan akurasi. (Hidayatullah, 2016)

Pada tabel 4.6 disajikan perbandingan data sebelum dan sesudah dilakukannya proses *Stopword Removal*.

Tabel 4.6 Contoh Data dengan *Stopword Removal*

sebelum	sesudah
bisa jadi inspirasi jangan masih muda udah malas kerja malah milih ngamen dan jadi tukang parkir	jadi inspirasi jangan masih muda udah malas kerja malah milih ngamen jadi tukang parkir
salut sama tukang parkir alfamidi cekatan dan ramah banget jadi keinget sama almarhum bapak	salut sama tukang parkir alfamidi cekatan ramah banget jadi keinget sama almarhum bapak
oalah anda tukang parkir di indomart saya ya minta yang suka minta duitnya maksa owkwkwk	oalah anda tukang parkir indomart saya ya minta suka minta duitnya maksa owkwkwk
cuma gara gara peluitnya tukang parkir aku bisa ngakak lho bahagia itu sederhana	cuma gara gara peluitnya tukang parkir aku ngakak lho bahagia sederhana
emang susah kok menghapus korupsi di negara ini orang dari tukang parkir aja ada yang doyan korupsi. apalagi petinggi	emang susah kok menghapus korupsi negara orang tukang parkir aja ada doyan korupsi. apalagi petinggi

4.3.5. Stemming

Proses *stemming* dilakukan untuk mengubah suatu kata menjadi bentuk akar kata (*stem*) proses ini dilakukan dengan cara menghilangkan imbuhan *prefix*, *infiks*, *suffiks*. Jika imbuhan tersebut tidak dihilangkan maka setiap satu kata dasar akan disimpan dengan imbuhan yang melekatinya sehingga hal tersebut akan menimbulkan *noise* (Apriandi, 2016)

Tabel 4.7 Contoh Data dengan *Stemming*

sebelum	sesudah
jadi inspirasi jangan masih muda udah malas kerja malah milih ngamen jadi tukang parkir	jadi inspirasi jangan muda udah malas kerja malah pilih ngamen jadi tukang parkir

Tabel 4.8 Contoh Data dengan *Stemming* (lanjutan)

sebelum	sesudah
salut sama tukang parkir alfamidi cekatan ramah banget jadi keinget sama almarhum bapak	salut sama tukang parkir alfamidi cekatan ramah banget jadi keinget sama almarhum bapak
oalah anda tukang parkir indomart saya ya minta suka minta duitnya maksa owkwkwk	oalah tukang parkir indomart ya minta suka minta duit maksa owkwkwk
cuma gara gara peluitnya tukang parkir aku ngakak lho bahagia sederhana	cuma gara gara peluit tukang parkir aku ngakak lho bahagia sederhana
emang susah kok menghapus korupsi negara orang tukang parkir aja ada doyan korupsi. apalagi petinggi	emang susah kok hapus korupsi negara orang tukang parkir ada doyan korupsi petinggi

4.4. Pembobotan Kata (Term Weighting)

Setelah melakukan pelabelan dan *preprocessing* data, akan dilakukan proses ekstraksi fitur. Dimana dalam penelitian ini menggunakan data text sejumlah 1357 data yang sudah melewati tahap pelabelan dan preprocessing data. Ekstraksi fitur berupa pembobotan kata (*term weighting*) dengan menggunakan metode *TF-IDF*. Data yang siap untuk dilakukan pembobotan ditampilkan pada tabel 4.8.

Tabel 4.9 Contoh Data untuk Pembobotan

tweet	label
salut sama tukang parkir alfamidi cekatan ramah banget jadi keinget sama almarhum bapak	2
oalah tukang parkir indomart minta suka minta duit maksa owkwkwk	1
cuma gara gara peluit tukang parkir aku ngakak lho bahagia sederhana	2
heran sama tukang parkir sini kerja enggak malak iya	1

4.4.1. Term Frequency – Invers Document Frequency (TF-IDF)

TF-IDF merupakan suatu metode yang digunakan untuk mencari bobot suatu kata dalam dokumen teks dengan rumus :

$$idf_t = \log \frac{N_d}{df_t} + 1 \quad (7)$$

N_d = Jumlah semua dokumen

df_t = Jumlah term yang muncul dalam keseluruhan dokumen

idf_t = Nilai sebaran term pada koleksi yang bersangkutan

Perhitungan *IDF* menggunakan persamaan (7) berdasarkan dokumen tweet pada tabel 4.9 ditunjukkan pada tabel sebagai berikut :

Tabel 4.10 Contoh Perhitungan IDF

term	TF				TF(Df)	IDF
	DT1	DT2	DT3	DT4		
aku	0	0	1	0	1	0.602059991
alfamidi	1	0	0	0	1	0.602059991
almarhum	1	0	0	0	1	0.602059991
bahagia	0	0	1	0	1	0.602059991
banget	1	0	0	0	1	0.602059991
bapak	1	0	0	0	1	0.602059991
cekatan	1	0	0	0	1	0.602059991
cuma	0	0	1	0	1	0.602059991
duit	0	1	0	0	1	0.602059991
enggak	0	0	0	1	1	0.602059991
gara	0	0	2	0	2	0.301029996
heran	0	0	0	1	1	0.602059991
indomart	0	1	0	0	1	0.602059991
lya	0	0	0	1	1	0.602059991
jadi	1	0	0	0	1	0.602059991
keinget	1	0	0	0	1	0.602059991

Tabel 4.11 Contoh Perhitungan *IDF* (lanjutan)

term	TF				TF(Df)	IDF
	DT1	DT2	DT3	DT4		
kerja	0	0	0	1	1	0.60206
Lho	0	0	1	0	1	0.60206
maksa	0	1	0	0	1	0.60206
malak	0	0	0	1	1	0.60206
minta	0	2	0	0	2	0.30103
ngakak	0	0	1	0	1	0.60206
oalah	0	1	0	0	1	0.60206
owkwkwkw	0	1	0	0	1	0.60206
parkir	1	1	1	1	4	0
peluit	0	0	1	0	1	0.60206
ramah	1	0	0	0	1	0.60206
salut	1	0	0	0	1	0.60206
sama	2	0	0	1	3	0.124939
sederhana	0	0	1	0	1	0.60206
sini	0	0	0	1	1	0.60206
suka	0	1	0	0	1	0.60206
tukang	1	1	1	1	4	0

Setelah mendapatkan nilai *IDF* pada masing masing kata, maka dilanjutkan proses menghitung bobot kata *TF-IDF* dengan menggunakan persamaan (5). Berdasarkan hasil *TF* dan *IDF* pada tabel 4.10 perhitungan *TF-IDF* ditunjukkan pada tabel sebagai berikut :

Tabel 4.12 Contoh Perhitungan *TF-IDF*

IDF	TF-IDF			
	WDT1	WDT2	WDT3	WDT4
0.602059991	0	0	0.60206	0
0.602059991	0.60206	0	0	0
0.602059991	0.60206	0	0	0
0.602059991	0	0	0.60206	0
0.602059991	0.60206	0	0	0
0.602059991	0.60206	0	0	0

Tabel 4.13 Contoh Perhitungan *TF-IDF* (lanjutan)

IDF	TF-IDF			
	WDT1	WDT2	WDT3	WDT4
0.60206	0.60206	0	0	0
0.60206	0	0	0.60206	0
0.60206	0	0.60206	0	0
0.60206	0	0	0	0.60206
0.30103	0	0	0.60206	0
0.60206	0	0	0	0.60206
0.60206	0	0.60206	0	0
0.60206	0	0	0	0.60206
0.60206	0.60206	0	0	0
0.60206	0.60206	0	0	0
0.60206	0	0	0	0.60206
0.60206	0	0	0.60206	0
0.60206	0	0.60206	0	0
0.60206	0	0	0	0.60206
0.30103	0	0.60206	0	0
0.60206	0	0	0.60206	0
0.60206	0	0.60206	0	0
0.60206	0	0.60206	0	0
0.60206	0	0.60206	0	0
0	0	0	0	0
0.60206	0	0	0.60206	0
0.60206	0.60206	0	0	0
0.60206	0.60206	0	0	0
0.124939	0.249877	0	0	0.124939
0.60206	0	0	0.60206	0
0.60206	0	0	0	0.60206
0.60206	0	0.60206	0	0
0	0	0	0	0

Setelah mendapatkan nilai *TF-IDF* pada masing masing kata, maka dilanjutkan proses klasifikasi dengan menggunakan metode *Naïve Bayes Classifier*.

4.5. Algoritma Naïve Bayes

Setelah diketahui bobot per kata dalam dokumen selanjutnya akan dilakukan proses klasifikasi menggunakan metode *Naïve Bayes*. Pada proses klasifikasi, mempunyai beberapa tahap diantaranya;

1. Training

- a. Menghitung probabilitas prior dengan persamaan (2).

$$P_{(positif)} = \frac{2}{4} = 0,5$$

$$P_{(negatif)} = \frac{2}{4} = 0,5$$

Keterangan :

P_l = probabilitas dokumen suatu label

N_l = jumlah label (l) pada dokumen

N = jumlah semua dokumen

- b. Menghitung probabilitas kata ke -n dengan persamaan (8).

$$P(t|l) = \frac{W_{lt} + 1}{(\sum w' \in vw'ct + B')} \quad (8)$$

Keterangan :

$\frac{t_n}{l}$ = probabilitas kata ke - n dari label l

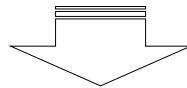
W_{lt} = nilai pembobotan *tfidf* atau W di label l

$\sum w' \in vw'ct$ = Jumlah total W dari keseluruhan data yang berlabel l

$B' = \text{Jumlah } W \text{ kata unik (nilai idf) pada seluruh dokumen}$

Tabel 4.14 Contoh Perhitungan TF-IDF (lanjutan)

term	TF				TF(Df)	IDF	TF-IDF			
	DT 1	DT 2	DT 3	DT 4			WDT 1	WDT 2	WDT 3	WDT 4
aku	0	0	1	0	1	$\frac{0.60205999}{1}$	0	0	$\frac{0.60206}{06}$	0
alfami di	1	0	0	0	1	$\frac{0.60205999}{1}$	$\frac{0.60206}{06}$	0	0	0
almarhum	1	0	0	0	1	$\frac{0.60205999}{1}$	$\frac{0.60206}{06}$	0	0	0
bahagi a	0	0	1	0	1	$\frac{0.60205999}{1}$	0	0	$\frac{0.60206}{06}$	0
banget	1	0	0	0	1	$\frac{0.60205999}{1}$	$\frac{0.60206}{06}$	0	0	0
bapak	1	0	0	0	1	$\frac{0.60205999}{1}$	$\frac{0.60206}{06}$	0	0	0
cekatan	1	0	0	0	1	$\frac{0.60205999}{1}$	$\frac{0.60206}{06}$	0	0	0



sederhana	0	0	1	0	1	0.60206	0	0	$\frac{0.60206}{06}$	0
sini	0	0	0	1	1	0.60206	0	0	0	$\frac{0.60206}{206}$
suka	0	1	0	0	1	0.60206	0	$\frac{0.60206}{06}$	0	0
tukang	1	1	1	1	4	0	0	0	0	0

Tabel 4.15 Contoh Perhitungan Probabilitas Laplace Smoothing

term	$\sum W \text{ kata } t$		IDF	Laplace smoothing	
	positif	negatif		P(positif)	P(negatif)
aku	0.60206	0	0.60206	0.057075	0.03916

Tabel 4.16 Contoh Perhitungan Probabilitas Laplace Smoothing (lanjutan)

term	$\sum W$ kata t		IDF	Laplace smoothing	
	positif	negatif		P(positif)	P(negatif)
alfamidi	0.60206	0	0.60206	0.05708	0.03916
almarhum	0.60206	0	0.60206	0.05708	0.03916
bahagia	0.60206	0	0.60206	0.05708	0.03916
banget	0.60206	0	0.60206	0.05708	0.03916
bapak	0.60206	0	0.60206	0.05708	0.03916
cekatan	0.60206	0	0.60206	0.05708	0.03916
cuma	0.60206	0	0.60206	0.05708	0.03916
duit	0	0.60206	0.60206	0.03563	0.06274
enggak	0	0.60206	0.60206	0.03563	0.06274
gara	0.60206	0	0.30103	0.05708	0.03916
heran	0	0.60206	0.60206	0.03563	0.06274
indomart	0	0.60206	0.60206	0.03563	0.06274
iya	0	0.60206	0.60206	0.03563	0.06274
jadi	0.60206	0	0.60206	0.05708	0.03916
keinget	0.60206	0	0.60206	0.05708	0.03916
kerja	0	0.60206	0.60206	0.03563	0.06274
lho	0.60206	0	0.60206	0.05708	0.03916
maksa	0	0.60206	0.60206	0.03563	0.06274
malak	0	0.60206	0.60206	0.03563	0.06274
minta	0	0.60206	0.30103	0.03563	0.06274
ngakak	0.60206	0	0.60206	0.05708	0.03916
oalah	0	0.60206	0.60206	0.03563	0.06274
owkwkwkw	0	0.60206	0.60206	0.03563	0.06274
parkir	0	0	0	0.03563	0.03916
peluit	0.60206	0	0.60206	0.05708	0.03916
ramah	0.60206	0	0.60206	0.05708	0.03916
salut	0.60206	0	0.60206	0.05708	0.03916
sama	0.24988	0.12494	0.12494	0.04453	0.04405
sederhana	0.60206	0	0.60206	0.05708	0.03916
sini	0	0.60206	0.60206	0.03563	0.06274
suka	0	0.60206	0.60206	0.03563	0.06274
tukang	0	0	0	0.03563	0.03916
Σ	10.4849	7.95172	17.5847		

$$\sum W(\text{positif}) = 10.4849$$

$$\sum W(\text{Negatif}) = 7.951719$$

$$\sum \text{IDF} = 17.58468$$

$$P(t|l) = \frac{W_{lt} + 1}{(\sum w' \in vw'ct + B')}$$

$$P(\text{bahagia}|\text{positif}) = \frac{0.6206 + 1}{(10.4849 + 17.58468)} = 0.057075$$

$$P(\text{bahagia}|\text{negatif}) = \frac{0 + 1}{(7.951719 + 17.58468)} = 0.057075$$

2. Testing

Setelah mendapatkan nilai dari probabilitas masing-masing kata terhadap setiap kelas atau label, maka selanjutnya dilakukan uji dengan data yang sama terhadap ketepatan klasifikasi secara manual. Proses pengujian menggunakan persamaan.

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n} P(t_k|c) \quad (9)$$

Untuk memudahkan penghitungan maka persamaan tersebut akan dihitung terlebih dahulu menggunakan tabel seperti dibawah ini. Pengujian klasifikasi dokumen negatif dengan kalimat “heran sama tukang parkir sini kerja engga malak iya” ditampilkan pada tabel 4.17.

Tabel 4.17 Contoh Perhitungan Kelas Negatif

term test	tf test	ls ^{tf} testing	
		positif	Negatif
Heran	1	0.035625761	0.062736336
Sama	1	0.044527836	0.044052367
Tukang	1	0.035625761	0.039159792
Parkir	1	0.035625761	0.039159792
sini	1	0.035625761	0.062736336
kerja	1	0.035625761	0.062736336
engga	1	0.035625761	0.062736336
malak	1	0.035625761	0.062736336
iya	1	0.035625761	0.062736336
nilai probabilitas		5.77717E-14	2.05937E-12
nilai max		2.05937E-12	

Untuk pengujian dokumen pertama dapat dilihat pada tabel diatas bahwa nilai probabilitas kelas dokumen negatif lebih besar daripada nilai probabilitas positif, jadi untuk pengujian pertama sudah sesuai. Selanjutnya pengujian klasifikasi dokumen positif dengan kalimat “cuma gara gara peluit tukang parkir aku ngakak lho bahagia sederhana” ditampilkan pada tabel berikut. Untuk *term* yang kemunculannya lebih dari satu kali, pangkatkan nilai *laplace smoothing* dengan jumlah kemunculan. Seperti pada term “gara” nilai probabilitas kelas dipangkatkan sesuai dengan jumlah kemunculannya.

Tabel 4.18 Contoh Perhitungan Kelas Positif

term test	tf test	ls ^{tf} testing	
		positif	negatif
Aku	1	0.057074606	0.03916
Bahagia	1	0.057074606	0.03916
Cuma	1	0.057074606	0.03916

Tabel 4.19 Contoh Perhitungan Kelas Positif (lanjutan)

term test	tf test	ls ^{tf} testing	
		positif	negatif
Gara	2	0.003258	0.001533
Lho	1	0.057075	0.03916
Ngakak	1	0.057075	0.03916
Parkir	1	0.035626	0.03916
Peluit	1	0.057075	0.03916
sederhana	1	0.057075	0.03916
Tukang	1	0.035626	0.03916
nilai probabilitas		4.08E-15	1.66E-16
nilai max		4.08E-15	

Untuk pengujian dokumen kedua dapat dilihat pada tabel diatas bahwa nilai probabilitas kelas dokumen positif lebih besar daripada nilai probabilitas negatif, jadi untuk pengujian kedua sudah sesuai.

Selanjutnya dilakukan proses pemodelan menggunakan klasifikasi *Naïve Bayes Classifier* dengan pembobotan *TF-IDF* terhadap data latih berjumlah 1357 data dengan komposisi label negatif sejumlah 760 data dan label positif sejumlah 597 data.

4.6. Evaluasi Model Klasifikasi

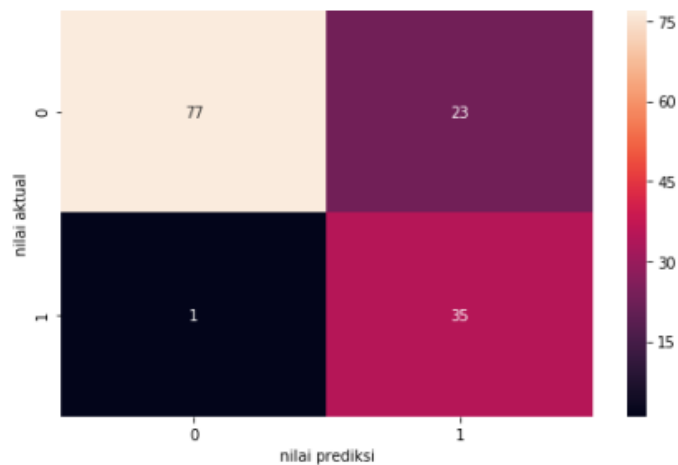
Proses klasifikasi yang telah dilakukan terhadap 1300 tweet dengan menggunakan metode klasifikasi *Naïve Bayes Classifier* perlu dilakukan evaluasi untuk menghitung akurasi dari model pembelajaran yang dibuat . Evaluasi model dalam penelitian ini menggunakan *split validation* untuk validasi dan *Confusion matrix* untuk menghitung performa model. *Split Validation* adalah teknik validasi yang membagi data menjadi dua bagian secara acak, sebagian sebagai data *training*

dan sebagian lainnya sebagai data *testing*. Dimana dalam penelitian ini *dataset* dengan jumlah 1357 tweet dibagi kedalam beberapa rasio pembagian yang selanjutnya akan diuji untuk melihat tingkat akurasinya masing-masing. Untuk skenario pengujian tertera pada tabel berikut.

Tabel 4.20 Skenario Pengujian *Split Validation*

Rasio	Jumlah Data	
	train	test
90:10	1221	136
80:20	1005	272
70:30	949	408

Hasil akurasi didapatkan dari perhitungan *confusion matrix* dari masing-masing pengujian, untuk pengujian pertama model klasifikasi yang dibuat dengan rasio data 90:10 menghasilkan prediksi sentimen positif 26,5%, dan 73,5% prediksi sentimen negatif. Grafik *confusion matrix* percobaan pertama ditampilkan dalam gambar 4.2.



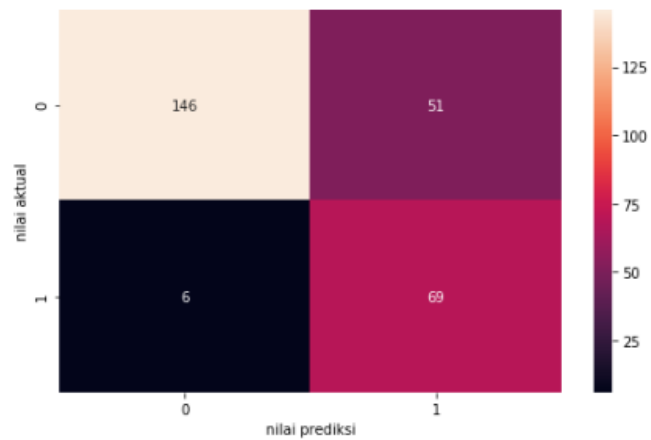
Gambar 4.2 Grafik *Confusion Matrix Percobaan Pertama*

Berdasarkan grafik *confusion matrix* diatas maka akurasi dari model klasifikasi dapat diukur dengan persamaan (6).

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

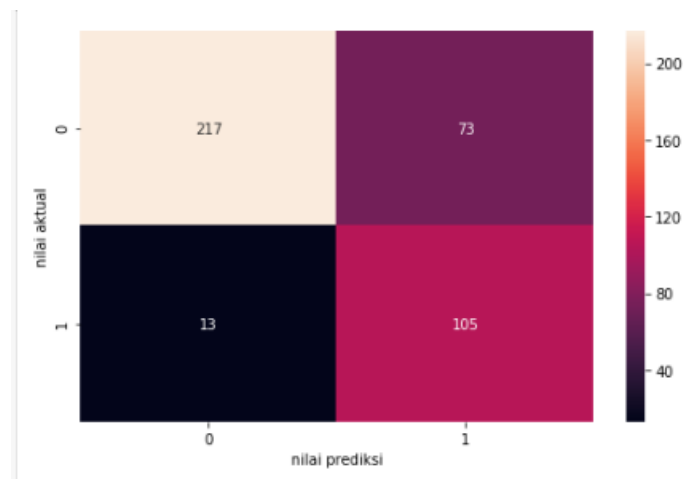
$$\frac{35 + 77}{35 + 1 + 77 + 23} = \frac{112}{136} = 0.8235 \times 100 = 82.35\%$$

Untuk pengujian kedua model klasifikasi yang dibuat dengan rasio data 80:20 menghasilkan prediksi sentimen positif 27,6%, dan 72,4% prediksi sentimen negatif. Grafik *confusion matrix* pengujian kedua ditampilkan dalam gambar 4.3.



Gambar 4.3 Grafik *Confusion Matrix* Percobaan Kedua

Untuk pengujian ketiga model klasifikasi yang dibuat dengan rasio data 70:30 menghasilkan prediksi sentimen positif 28,9%, dan 71.1% prediksi sentimen negatif. Grafik *confusion matrix* percobaan ketiga ditampilkan dalam gambar berikut :



Gambar 4.4 Grafik *Confusion Matrix* Percobaan Ketiga

Adapun hasil akurasi yang diperoleh dari masing-masing pengujian berdasarkan tabel 4.14 ditunjukkan pada tabel berikut :

Tabel 4.21 Hasil Pengujian *Split Validation*

Rasio		hasil akurasi
Train	test	
90	10	82.35%
80	20	79.04%
70	30	78.92%

Berdasarkan hasil dari tiga pengujian diatas, dapat dilihat model klasifikasi yang dibuat dengan rasio data 90:10 menghasilkan akurasi 82,35%, model klasifikasi yang dibuat dengan rasio 80:20 menghasilkan akurasi 79,04%, dan model klasifikasi yang dibuat dengan rasio 70:30 menghasilkan akurasi 7,92%. Dapat dilihat bahwa akurasi paling tinggi dari ketiga model tersebut diperoleh dari model klasifikasi yang dibuat dengan rasio data 90:10.

4.7. Implementasi

Pada tahap implementasi ini dilakukan klasifikasi secara otomatis menggunakan model klasifikasi yang sudah dibuat sebelumnya dengan menggunakan metode klasifikasi *Naïve Bayes* dan *TF-IDF* sebagai pembobotan kata. Model pembelajaran dengan akurasi tertinggi digunakan untuk klasifikasi terhadap data baru yang belum memiliki label. Data baru tersebut berupa tweet yang diambil pada mulai tanggal 8 Juli 2020 sampai 12 Juli 2020 sebanyak 1000 data . Setelah dilabeli secara otomatis, maka didapatkan hasil klasifikasi sentimen positif sebanyak 352 tweet dan negatif sebanyak 648 tweet. Presentase kelas hasil prediksi ditampilkan dalam gambat 4.5

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan data yang diambil dari twitter sebanyak 5017 data, kemudian dilakukan proses seleksi data dengan mencari nilai modus dari pelabelan secara manual oleh 3 orang yang berbeda dan membuang twit yang tidak sesuai, menghasilkan 1357 data dengan 597 tweet berlabel positif dan 760 tweet berlabel negatif. Data diolah menggunakan metode *Naïve Bayes Classifier* menggunakan pembobotan *TF-IDF* serta *split validation* dan confusion matrix untuk membangun model klasifikasi terbaik.

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa *Naïve Bayes Classifier* dapat digunakan untuk mengklasifikasi data tweet dengan kata kunci Tukang Parkir. Rasio *split validation* yang digunakan pada masing masing model menunjukkan akurasi yang berbeda dengan menggunakan evaluasi *confusion matrix* model dengan rasio *split* 90:10 menghasilkan akurasi 82,35%, rasio *split* 80:20 menghasilkan akurasi 79,04%, sedangkan rasio *split* 70:30 menghasilkan akurasi 78,92%. Dapat dilihat model dengan rasio *split* 90:10 mempunyai performa akurasi tertinggi, untuk selanjutnya akan digunakan untuk implementasi terhadap data baru. Hasil implementasi pada data uji sejumlah 1000 tweet yang diambil pada rentang waktu 7-12 Juli menggunakan model yang sudah dibuat menghasilkan prediksi tweet positif sebanyak 28,9% dan prediksi tweet negatif sebanyak 71,1%.

5.2. Saran

Pada penelitian ini, peneliti tentu tidak jauh dari kekurangan. Oleh karena itu, terdapat beberapa saran untuk dijadikan perbaikan untuk penelitian selanjutnya.

Saran-saran tersebut meliputi :

1. Pemilihan *dataset* yang baik untuk membangun model klasifikasi dengan performa yang lebih baik
2. Membandingkan kembali dengan metode pembelajaran mesin lainnya sehingga dapat meningkatkan hasil pengklasifikasian pada data objek yang diteliti.
3. Merancang Sistem Sentimen Analisis yang dapat langsung digunakan secara praktis

DAFTAR PUSTAKA

- Akbari, M., Novianty, A., & Setianingsih, C. (2017). Analisis Sentimen Menggunakan Metode Learning Vector Quantization. *E-Proceeding Of Engineering*, Vol.4, No.2 Agustus(ISSN : 2355-9365).
- Annur, H. (2018). Klasifikasi Masyarakat Miskin Menggunakan Metode Naïve Bayes. *ILKOM Jurnal Ilmiah*, Vol 10(e-ISSN 2548-7779).
- Apriandi, D., Afandi, M., & Wahyuni, E. (2016). Analisis Sentimen Pelanggan Wifi.Id Pada Twitter Dengan Support Vector Machine. *Jurnal Sistem Informasi Dan Bisnis Cerdas (SIBC)*, Vol. 9, No. 1.
- Gorunescu, F. (2011). *Data mining Concepts, Models and Techniques*. Verlag Berlin Heidelberg: Springer.
- Hamidah, N. (2019). Analisis Sentimen Masyarakat Terhadap Fintech Menggunakan Metode Naïve Bayes Classifier.
- Han, J., Kamber, M., & Pei, J. (2012). *Data mining*. Amsterdam: Elsevier/Morgan Kaufmann.
- Hidayatullah, A. (2016). Pengaruh Stopword Terhadap Performa Klasifikasi Tweet Berbahasa Indonesia. *JISka*, Vol. 1, No. 1 (ISSN 2527-5836).
- Hidayatullah, A., & SN, A. (2014). Analisis Sentimen Dan Klasifikasi Kategori Terhadap Tokoh Publik Pada Twitter. *Seminar Nasional Informatika 2014 (Semnasif 2014) UPN "Veteran" Yogyakarta*, (ISSN: 1979-2328).
- Juru parkir. Id.wikipedia.org. (2020). Retrieved 14 July 2020, from https://id.wikipedia.org/wiki/Juru_parkir.
- Kursini, Luthfi, E. T. (2009). *Algoritma Data Mining*, Andi Offset, Yogyakarta.
- Liu, B. (2012). *Sentimen Analysis and Opinion Mining*. Chicago: Claypool Publishers.
- Lorosae, T., Prakoso, B., & Kusrini, S. (2020). Analisis Sentimen Berdasarkan Opini Masyarakat Pada Twitter Menggunakan Naïve Bayes. *Seminar Nasional Teknologi Informasi Dan Multimedia 2018*, (ISSN : 2302-3805).

- Luqyana, W., Cholissodin, I., & Pradana, R. (2018). Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, Vol. 2, No. 11(e-ISSN: 2548-964X).
- Mahardika, Y., & Zuliarso, E. (2018). Analisis Sentimen Terhadap Pemerintahan Joko Widodo Pada Media Sosial Twitter Menggunakan Algoritma Naïves Bayes Classifier. *Prosiding Sintak*, (ISBN: 978-602-8557-20-7).
- Pintoko, B., & L, K. (2018). Analisis Sentimen Jasa Transportasi Online pada Twitter Menggunakan Metode Naïve Bayes Classifier. *E-Proceeding Of Engineering*, Vol.5, No.3 (ISSN : 2355-9365), 8121.
- Pramita, D., Saptono, R., & Anggrainingsih, R. (2018). Academic Articles Classification Using Naïve Bayes Classifier (NBC) Method. *ITSMART: Jurnal Ilmiah Teknologi Dan Informasi*, Vol. 7, No. 2, December 2018 (ISSN: 2301-7201).
- Rapid Miner. (2014). *Operator Refence Manual*. Boston: Rapid Miner Inc.
- Smetanin, S. (2020). Sentiment Analysis of Tweets using Multinomial Naïve Bayes. *towardsdatascience.com*. Retrieved 15 July 2020, from <https://towardsdatascience.com/sentiment-analysis-of-tweets-using-multinomial-naïve-bayes-1009ed24276b>.
- Suryono, S., Utami, E., & Luthfi, E. T. (2018). Klasifikasi Sentimen Pada Twitter Dengan Naïve Bayes Classifier. *Angkasa: Jurnal Ilmiah Bidang Teknologi*, 10 (1), 8996.

LAMPIRAN

Lampiran 1 Source Code Crawling API

```
import tweepy
from tweepy import OAuthHandler
import pandas as pd
consumer_key = "XXX"
consumer_secret = "XXX"
access_token = "XXX"
access_secret = "XXX"
auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)
api = tweepy.API(auth, wait_on_rate_limit=True)

search_words = 'tukang parkir since:2020-07-08 until:2020-07-12 -
filter:retweets'

twit = tweepy.Cursor(api.search, q = search_words, lang='id',
tweet_mode='extended').items(1000)

fieldnames = []
for status in twit:
    fieldnames.append({'username' : status.user.screen_name,
                      'tweet' : status.full_text})

tweet = pd.DataFrame(fieldnames, columns=['username', 'tweet'])
tweet

tweet.to_csv('tukang parkir 8-10 julyy.csv', encoding='utf-8',
sep=',', index=False)
```

Lampiran 2 Source Code Preprocessing

```
import pandas as pd

data = pd.read_csv("2label.csv")

data.head()

CLEANSING

import re

import string

from tqdm.notebook import tqdm

output_text = []

def clean_text(list_of_text):

    for text in tqdm(list_of_text):

        text = re.sub(r"(?:\@|https?://)\S+", "", text)

        text = re.sub(r"[^a-zA-Z0-9]", ' ', text)

        output_text.append(text)

    return output_text

data['text'] = clean_text(list(data['text']))

data.to_csv("data_cleansing.csv", index=False)

SLANGWORD CONVERTER

data_cleansing = pd.read_csv("data_cleansing.csv")
```

Lampiran 2 Source Code Preprocessing

```

slangword = pd.read_csv("Slangword-indonesian.csv")

import nltk

def slang(text):
    text = str(text).lower()
    text = nltk.word_tokenize(text)
    ul = 0
    for a in slangword['slang']:
        w = 0
        for kt in text:
            if kt ==a:
                text[w] = slangword['formal'][ul]
                w = w+1
            ul=ul+1
        kal = ""
        for kl in text:
            kal = kal + " " + kl
        text = kal
    return kal

data_cleansing['text'] = data_cleansing['text'].apply(lambda text : slang(text))

data_cleansing.to_csv("data_noslang.csv", index=False)

STOPWORD

data_noslang = pd.read_csv("data_noslang.csv")

```

Lampiran 2 Source Code Preprocessing

```

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory

factory = StopWordRemoverFactory()

stopword = factory.create_stop_word_remover()

output_text = []

def stop(list_of_text):

    for text in tqdm(list_of_text):

        text = stopword.remove(text)

        output_text.append(text)

    return output_text

data_noslang['text'] = stop(list(data_noslang['text']))

data_noslang.to_csv("data_stop.csv", index=False)


STEMMING

data_stop = pd.read_csv("data_stop.csv")

# create stemmer

stem_factory = StemmerFactory()

stemmer = stem_factory.create_stemmer()

output_text = []

```

Lampiran 2 Source Code Preprocessing

```
def stem_data(list_of_text):  
  
    for text in tqdm(list_of_text):  
  
        text = stemmer.stem(text)  
  
        output_text.append(text)  
  
    return output_text  
  
data_stop['text'] = stem_data(list(data_stop['text']))  
  
data_stop.to_csv("data_clean.csv",index=False)
```


Lampiran 3 Source Code Membangun Model

```
import pandas as pd

data = pd.read_csv("data_clean.csv")

data.head()

#Split Data

from sklearn.model_selection import train_test_split

tweet = data.text

label = data.label

tweet_train, tweet_test, label_train, label_test =

train_test_split(tweet,label,test_size=0.1, random_state=0)

#TF-IDF Vectorizer

from sklearn.feature_extraction.text import TfidfVectorizer

tfvec = TfidfVectorizer()

#NaïveBayes

from sklearn.Naïve_bayes import MultinomialNB

from sklearn.pipeline import Pipeline

clf_mnb = MultinomialNB()

model_mnb = Pipeline([("TF-IDF", tfvec),

                        ('classifier', clf_mnb)])

model_mnb.fit(tweet_train, label_train)

hasil_mnb = model_mnb.predict(tweet_test)

#Confusion Matrix
```

Lampiran 3 Source Code Membangun Model

```

from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

confusion_matrix(hasil_mnb, label_test)

import seaborn as sns

import matplotlib.pyplot as plt

f, ax = plt.subplots(figsize=(8, 5))

sns.heatmap(confusion_matrix(hasil_mnb, label_test), annot=True, fmt=".0f",
ax=ax)

plt.xlabel("nilai prediksi")

plt.ylabel("nilai aktual")

plt.show()

#Convert hasil ke list (hasil tadinya dalam bentuk ndarray)

l_hasil = list(hasil_mnb)

print("Prediksi tweet baik: " + str(l_hasil.count(2)))

print("Prediksi tweet kurang baik: " + str(l_hasil.count(1)))

num = 100*(l_hasil.count(2)/tweet_test.count())

print("Persentase tweet baik = {:.2f} %".format(num))

acc = 100*accuracy_score(hasil_mnb,label_test)

print("Ketelitian = {:.2f} %".format(acc))

```

Lampiran 3 Source Code Membangun Model

```

import matplotlib.pyplot as plt
sizes = [l_hasil.count(2), l_hasil.count(1)]
labels = ['positive', 'negative']
colors = ['blue', 'red']
explode = (0.1, 0)
fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='% 1.1f%%',
        shadow=True, startangle=90, colors=colors)
ax1.axis('equal')
plt.legend(labels, loc="best")
plt.show()

#Memngekspor Modul
import pickle
# Nama fike tempat menyimpan data
model = 'model_klasifikasi.data'
# Siapkan file untuk ditulis
f = open(model, 'wb')
# Simpan data nama siswa ke dalam file dengan perintah dump
pickle.dump(model_mnb, f)
f.close() # tutup setelah file digunakan

```

Lampiran 4 Source Code Implementasi

```
import pickle
model_mnb = 'model_klasifikasi.data'
# buka file namasiswa.data
f = open(model_mnb, 'rb')
# load data dari file tersebut
model_mnb = pickle.load(f)
f.close()
print(model_mnb)
import pandas as pd
data_test = pd.read_csv("tukang parkir 8-10 july.csv")
data_test.head()
y = pd.DataFrame(columns=['text','label'])
hasil_tes = model_mnb.predict(data_test.tweet)
y['label'] = hasil_tes
y['text'] = data_test['tweet']
y.head()
l_hasil = list(hasil_tes)
print("Prediksi tweet baik: " + str(l_hasil.count(2)))
print("Prediksi tweet kurang baik: " + str(l_hasil.count(1)))
#Grafik Piechart
import matplotlib.pyplot as plt
sizes = [l_hasil.count(2), l_hasil.count(1)]
labels = ['positive','negative']
colors = ['blue', 'red']
explode = (0.1,0)
```

Lampiran 4 Source Code Implementasi

```
#patches, texts = plt.pie(sizes, colors=colors, startangle=-270)

fig1, ax1 = plt.subplots()

ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90, colors=colors)

ax1.axis('equal')

plt.legend(labels, loc="best")

plt.show()

label_positif = y[y['label'] == 2]

from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt

text = label_positif.text.values

wordcloud = WordCloud(
    width = 3000,
    height = 2000,
    background_color = 'black',
    stopwords = STOPWORDS).generate(str(text))

fig = plt.figure(
    figsize = (40, 30),
    facecolor = 'k',
    edgecolor = 'k')

plt.imshow(wordcloud, interpolation = 'bilinear')

plt.axis('off')

plt.tight_layout(pad=0)

plt.show()

label_negatif = y[y['label'] == 1]
```

Lampiran 4 Source Code Implementasi

```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
text = label_negatif.text.values
wordcloud = WordCloud(
    width = 3000,
    height = 2000,
    background_color = 'black',
    stopwords = STOPWORDS).generate(str(text))
fig = plt.figure(
    figsize = (40, 30),
    facecolor = 'k',
    edgecolor = 'k')
plt.imshow(wordcloud, interpolation = 'bilinear')
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

CURRICULUM VITAE

A. Biodata Diri

Nama Lengkap : Muhammad Sholihuddin Nur

Jenis Kelamin : laki-laki

Tempat, Tanggal Lahir : Wonogiri, 20 September 1998

Alamat Asal : Jl Nanas I, Gerdu, Giripurwo,
Wonogiri

Alamat Tinggal : Jl Sunten, Banguntapan, Bantul

E-mail : sholihuddinnur@gmail.com



B. Latar Belakang Pendidikan Formal

Jenjang	Nama Sekolah	Tahun
SD	SD NEGERI 1 WONOGIRI	2004 – 2010
SMP	SMP NEGERI 1 WONOGIRI	2010 – 2013
SMA	SMA NEGERI 2 WONOGIRI	2013 – 2016
S1	UIN SUNAN KALIJAGA YOGYAKARTA	2016 – 2020