

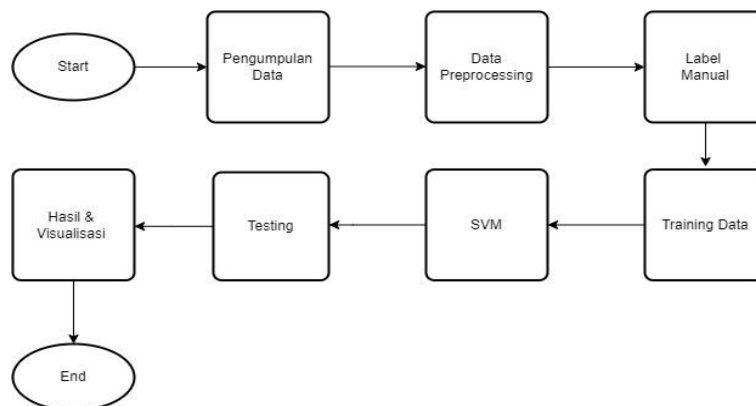
BAB 3 METODE PENELITIAN

Penelitian ini adalah penelitian analisis sentimen positif dan negatif pada data Twitter. Metode *Support Vector Machine (SVM)* digunakan pada penelitian ini. Data *tweet* yang didapatkan dari Twitter yang berkaitan dengan aplikasi bimbingan belajar *online*, kemudian dilakukan tahap pengolahan data berupa *preprocessing* untuk mendapatkan hasil yang diinginkan. Data tersebut nantinya digunakan untuk memetakan informasi atau sentimen dari *netizen* di Twitter mengenai aplikasi bimbingan belajar online sehingga didapatkan informasi yang sesuai dengan kondisi mengenai aplikasi-aplikasi bimbingan belajar *online* saat ini.

Penelitian ini diawali dengan latar belakang permasalahan yang ada, mengolah data yang sudah didapatkan kemudian menentukan sentimen yang tepat sehingga diperoleh informasi yang sesuai dengan apa yang diharapkan. Dibawah ini adalah alur, bahan, alat, dan jalannya penelitian analisis sentimen aplikasi bimbingan belajar *online* serta tahapan penelitian guna menyelesaikan proses analisis sentimen menggunakan data *tweet*.

3.1 ALUR PENELITIAN

Alur penelitian yang dilakukan , dapat dilihat pada Gambar 3.1



Gambar 3.1 Alur penelitian

1. Pengumpulan Data

Proses pengumpulan data menggunakan crawling API Twitter dengan menggunakan modul *tweepy* pada bahasa python. Data yang diambil memiliki kata kunci yang bertema aplikasi bimbingan belajar *online* seperti *ruangguru*, *quipper*, *zenius*, *rumahbelajar*, dan *cakap*.

2. *Text Preprocessing*

Setelah proses pengumpulan data selanjutnya dilakukan *text preprocessing* untuk menghilangkan serta mengatasi *noisy* agar digunakan secara maksimal. Adapun alur proses *text preprocessing* antara lain, *case folding*, *filtering*, *tokenizing*, dan *stopword removal*. Setelah data bersih maka dilakukan proses pelabelan secara manual.

3. *Vektorisasi*

Penelitian ini menerapkan algoritma pembobotan kata atau *TF-IDF* (*Term Frequency-Inverse Document Frequency*) yang berguna untuk melakukan pembobotan sebagai cara untuk memberi nilai pada kata (*term*) berdasarkan suatu dokumen/*tweet*.

4. *Support Vector Machine*

Pada tahap ini terlebih dahulu dilakukan proses pembagian data menjadi 2 yaitu 80% data latih dan 20% data uji. Data latih digunakan untuk melatih algoritma dan pengembangan pemodelan, sedangkan data uji digunakan sebagai acuan dan untuk mengetahui performa dari algoritma atau model yang sudah dilatih sebelumnya. Selanjutnya barulah proses pemodelan dengan metode SVM.

5. Evaluasi

Tahap evaluasi dilakukan untuk mengetahui keakuratan dari pemodelan yang telah diterapkan pada data latih. Kemudian membandingkan hasil dari dua dataset yang berbeda dengan menerapkan *confusion matrix* untuk menghitung *precision*, *recall*, *f1-score*, dan *accuracy*.

6. Visualisasi

Pada tahap ini memberikan hasil terkait rekomendasi kepada pengguna terutama siswa terhadap aplikasi bimbingan belajar *online* dengan rating baik yang disajikan di dalam *dashboard* berbasis *webite application*.

3.2 BAHAN PENELITIAN

Bahan yang akan dibutuhkan untuk penelitian ini yaitu data *tweet*, *re-tweet* maupun komentar di Twitter yang berkaitan dengan aplikasi-aplikasi bimbingan belajar *online* seperti Ruangguru, Quipper, Zenius, Rumahbelajar, dan Cakap.

3.3 ALAT PENELITIAN

Alat yang digunakan dalam penelitian ini adalah komputer dengan spesifikasi yang memadai untuk menjalankan sistem operasi, pengembangan perangkat lunak, dan konektivitas internet. Sistem Operasi dan program-program aplikasi yang dipergunakan dalam pengembangan aplikasi ini adalah:

1. Sistem Operasi: Windows 10 64-bit.
2. Bahasa Pemrograman Python 3.7.4.
3. Microsoft Office Excel 2010
4. Jupyter Notebook 6.0.1

Library yang digunakan dalam penelitian ini adalah sebagai berikut :

- a. *Sklearn* : *Library* berbasis Python untuk membangun model pembelajaran mesin.
- b. *Library pandas* : Berfungsi untuk manipulasi serta pembersihan data.
- c. *Numpy* : Digunakan untuk mendukung proses komputasi atau perhitungan *numeric*.
- d. *Nltk* : Digunakan untuk mendukung proses pengolahan bahasa natural pada data teks agar mempermudah dalam proses *classification*, *tokenization* dan *stemming*.

3.4 JALAN PENELITIAN

Metode *Support Vector Machine (SVM)* digunakan dalam penelitian ini, metode ini dipilih karena dari beberapa penelitian metode *SVM* dapat melakukan

proses komputasi secara cepat dan memiliki tingkat akurasi lebih tinggi daripada metode lain. Penelitian ini menggunakan sosial media Twitter untuk proses pengambilan data.

Pada tahap awal penelitian ini dilakukan identifikasi dan pengumpulan data *tweet* dengan teknik *scrapping* dengan bantuan *tools* jupyter notebook, dimana proses ini menggunakan bahasa pemrograman python. Kriteria data yang diambil menggunakan *keyword* atau *hashtag* yang berhubungan dengan #ruangguru, #quipper , #Zenius, #rumahbelajar dan #cakap. Data yang berhasil diambil akan disimpan dalam format *Comma Separated Value (CSV)* atau dengan format excel. Setelah data terkumpul selanjutnya dilakukan tahap *preprocessing*.

Pada tahap ini dilakukan :

1. *filtering duplicate* = tahapan pertama dimana isi dari *tweet* yang sama akan dihapus untuk menghindari duplikat.
2. *case folding* = proses mengubah semua huruf menjadi huruf kecil di dalam dokumen.
3. *Cleaning* = proses menghapus karakter selain huruf seperti URL, mention dan hashtag.
4. *Tokenizing* = tahap pemisahan string input berdasarkan tiap kata.
5. *Filtering* = tahapan pengambilan kata-kata penting dari hasil *tokenizing*.

Pada langkah selanjutnya dilakukan tahapan pengelompokan data yang berguna untuk *clustering*. Selanjutnya pada data yang sudah terbagi kelompok *clustering* metode *Support Vector Machine (SVM)* digunakan untuk tahap analisis data, tahap ini dilakukan untuk mendapatkan data opini positif, negatif, dan netral.

Pada tahap terakhir yaitu tahap visualisasi, tahap ini memberikan hasil terkait rekomendasi kepada pengguna terutama siswa terhadap aplikasi bimbingan belajar online dengan rating baik yang disajikan di dalam dashboard berbasis *webite application*. Website dibuat menggunakan flask dengan bahasa pemrograman *python*. Adapun fitur-fitur yang tersedia didalam aplikasi *website* yang akan dikembangkan antara lain :

- Menu *Dashboard* = berisi ringkasan dari pencarian *keyword*, berapa mention yang tersedia, sumber pengambilan data, gambaran grafik tanggal *tweet*, *word cloud*, berapa jumlah positif, negatif, dan netral, profile username yang melakukan *tweet* dan lokasi user.
- Menu *Report* = berisi tentang laporan *keyword* berdasarkan harian, bulanan, dan lokasi user.
- Menu *User*
- Menu *Import / Export*

3.4.1 Pengambilan Data

Pengambilan Data adalah tahap dalam pengambilan data *tweet* dari mengenai topik ulasan layanan mengenai bimbingan belajar *online* meliputi ruangguru, zenius, quipper, rumah belajar, dan cakap dengan menggunakan Jupyter Notebook . Data *tweet* yang diambil menggunakan kata kunci ”#ruangguru, #zenius, #quipper, #rumah belajar, dan #cakap” dengan rentan waktu 1 Januari – 30 juni 2020 dengan jumlah *tweet* yang didapatkan adalah 24.311 dengan campuran data *tweet* dan *re-tweet* yang dimana akan ditampilkan di Microsoft Office Excel dalam format CSV. Pada Gambar 3.2 merupakan source code untuk pengambilan data.

```
# Setting variables to be used below
maxTweets = 5000

# Creating list to append tweet data to
tweets_list2 = []

# Using TwitterSearchScrapper to scrape data and append tweets to list
for i,tweet in enumerate(sntwitter.TwitterSearchScrapper('aplikasi Ruang Guru since:2020-01-01 until:2020-06-30 lang:id').get_items()):
    if i>maxTweets:
        break
    tweets_list2.append([tweet.date, tweet.content, tweet.username])
```

Gambar 3.2 Source code pengambilan data

Data yang tersimpan dari proses pengambilan data ini adalah data *tweet* meliputi tanggal, *username*, dan *tweet* yang dipisahkan oleh titik koma pada setiap kolomnya. Terdapat berbagai macam bentuk variasi yang ditampilkan mulai dari *username*, angka, *hashtag*, link URL yang terdapat didalam teks *tweet* dan *re-tweet* yang nantinya datanya perlu diolah kembali agar mendapatkan sesuai

dengan keinginan. Berikut contoh data *tweet* yang sudah dilakukan pengambilannya pada Tabel 3.1

Tabel 3.1 Contoh data *tweet* dan *re-tweet*

No	Tweet
1	@ahmadzakiyudin_ @zeniuseducation Pembahasan yang grand to sesi 1 nya gak ada??
2	@zeniuseducation Min kalau kayak gini kenapa ya? udah pake data, wi-fi, update apps tetep kayak gini soalnya https://t.co/nNJoFeYXyk
3	/edc di quipper selain ppu subtest apa lagi yg bagus? sebutin 1 aja
4	@subsbmptnfess buka di quipper sama zenius dek lumayan buat basic menurutku
5	@syubidudipa @sbmptnfess di quipper itu aksesnya gratis atau berbayar?
6	@sayuhuu @ardfadhil Langganan ruangguru aja pdhl lebih murah
7	Daftar ruangguru, zenius atau quipper aja https://t.co/uNw1mVsZWT
8	@halonitaa Ini sesi tanya jawab ruang guru ya
9	Pengguna Aplikasi Cakap Naik 5 Kali Selama Pandemi https://t.co/VhXcuzniGN
10	4 gb untuk akses aplikasi cakap. Yang termasuk aplikasi cakap apa aja ya min? @Telkomsel https://t.co/hVx5eeKtXG

3.4.2 Pre-processing Data

Pre-processing menjadi tahapan awal klasifikasi teks untuk menyiapkan data teks agar dapat digunakan nantinya pada proses yang berikutnya dan membuat teks menjadi sebuah informasi dengan kualitas yang baik. Proses dari tahapan *pre-processing* ini memerlukan beberapa library dari python untuk membantu dalam proses *pre-processing* yaitu *Library pandas* yang memiliki fungsi untuk manipulasi serta pembersihan data, *numpy* untuk mendukung proses komputasi atau perhitungan numerik, *nlTK* untuk mendukung proses pengolahan bahasa natural pada data teks agar mempermudah dalam proses *classification*, *tokenization* dan *stemming*, Berikut *import library* yang perlu dilakukan sebelum melakukan *pre-processing* dapat dilihat pada Gambar 3.3

```
import pandas as pd, numpy as np, nltk, string, re
from pandas import DataFrame
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

Gambar 3.3 Library preprocessing

Dan untuk pemanggilan data *tweet* yang terdapat pada file CSV akan di panggil ke dalam data frame untuk mempermudah dalam menjalankan proses *pre-processing*. Berikut kode pemanggilan ke dalam data frame terdapat pada Gambar 3.4`

```
def load_data():
    data = pd.read_csv('RumahBelajar.csv', nrows=None, names=['DateTime', 'Tweet ID', 'Text', 'Username'])
    return data

data = load_data()

data.head()
```

Gambar 3.4 Source code load data

Berikut merupakan tahapan-tahapan pada *pre-processing* data :

1. Cleaning Data

Cleaning data berfungsi untuk menghapus link url, simbol, angka, hastag. Berikut merupakan source code *cleaning* data. Berikut merupakan source code *cleaning* data dapat dilihat pada Gambar 3.5

```
def url_remove(tweet):
    t = re.sub(r'http\S+', '', tweet)
    return t

def punc_remove(tweet):
    t = re.sub(r'^\W\s', '', tweet)
    return t

def rt_remove(tweet):
    t = re.sub(r'RT[\s]+', '', tweet)
    return t

def number_remove(tweet):
    t = re.sub(r'[0-9]+', '', tweet)
    return t
```

Gambar 3.5 Source code cleaning data

Setelah menjalankan source code *cleaning* data, didapatkan data yang sudah bersih adapun perbandingan data sebelum *cleaning* dan data sesudah *cleaning* dapat dilihat pada Tabel 3.2

Tabel 3.2 Data sebelum *cleaning*

No	Data Tweet Kotor
1	@preciousseul malaskuu buka quipper,, coba buka ko dibagian intensif bru di persiapan, bagian pbm nya
2	@awbitious Gaess numpang nanya ya, itu quipper udah fix bgt sampe besok aja ya?
3	@ambitiousneithh @EdukaSystem halo mau tanya, to quipper itu di aplikasi kan? tapi saya udah cari kok gaada section buat ngerjain to ya. kalau boleh tau di bagian mana? makasih
4	@kejaibantuan Ditambah dengan quipper, aduh parah!
5	@ruangguru iseng banget pengen belajar bahasa inggris, tapi ruang guru hadir untuk pelajaran sekolah saja, sehingga ya tidak terlalu advance but it's okay.
6	@Ini_Talkshow @ruangguru Wahh terima kasih setelah ponakan saya Download ruang guru, dia lebih aktif lagi main game online nyaa..
7	@zeniuseducation @JeromePolin Videonya bisa di putar lagi ga ka?
8	@zeniuseducation Mau nanya serius ini min atau yg lain jawab flish, kalo udh premium paket ini dapet fitur apa lagi sih selain to? Ko gue gapaham yaa menangis diri ini 😭
9	@Asknonym kalo sepedaannya ga salah sih mnrt w. asal protokol kesehatannya tetep harus diperhatiin bener2. krn w mikirnya kita juga ga tau ini pandemi kapan selesai. jadi mungkin harus bener2 belajar "hidup normal baru", gabisa diem di rumah sampe kiamat :(
10	@TokopediaCare saya telah membeli paket pelatihan di tokopedia, dan sudah mendapatkan sertifikat. Setelah itu saya mereview pembelian, dan aplikasi cakap (tempat pelatihan), namun hingga saat ini sertifikat tidak ada di dashboard prakerja saya. Email gedeandi8@gmail.com

Setelah dilakukan *cleaning* data, maka data akan tampil seperti tabel 3.3 seperti berikut :

Tabel 3.3 Data sesudah *cleaning*

No	Data Tweet Kotor
1	preciousseul malaskuu buka quipper, coba buka ko dibagian intensif bru di persiapan, bagian pbm nya
2	awbitious Gaess numpang nanya ya, itu quipper udah fix bgt sampe besok aja ya
3	ambitiousneithh EdukaSystem halo mau tanya, to quipper itu di aplikasi kan tapi saya udah cari kok gaada section buat ngerjain to ya. kalau boleh tau di bagian mana makasih
4	kejaibantuan Ditambah dengam quipper, aduh parah
5	ruangguru iseng banget pengen belajar bahasa inggris, tapi ruang guru hadir untuk pelajaran sekolah saja, sehingga ya tidak terlalu advance but it's okay.
6	Ini_Talkshow ruangguru Wahh terima kasih setelah ponakan saya Download ruang guru, dia lebih aktif lagi main game online nyaa..
7	zeniuseducation JeromePolin Videonya bisa di putar lagi ga ka
8	zeniuseducation Mau nanya serius ini min atau yg lain jawab flish, kalo udh premium paket ini dapet fitur apa lagi sih selain to Ko gue gapaham yaa menangis diri ini
9	Askronym kalo sepedaannya ga salah sih mnrt w. asal protokol kesehatannya tetep harus diperhatiin bener2. krn w mikirnya kita juga ga tau ini pandemi kapan selesai. jadi mungkin harus bener2 belajar "hidup normal baru", gabisa diem di rumah sampe kiamat
10	TokopediaCare saya telah membeli paket pelatihan di tokopedia, dan sudah mendapatkan sertifikat. Setelah itu saya mereview pembelian, dan aplikasi cakap tempat pelatihan, namun hingga saat ini sertifikat tidak ada di dashboard prakerja saya. Email gedeandigmail.com

2. Tokenization

Tokenization berfungsi untuk merubah kalimat menjadi potongan kata. Berikut source code pada *tokenization* yang ditampilkan pada Gambar 3.6

```
cleaned = []

def clean_text(tweet):
    for i in tweet:
        cleaned.append(url_remove(punc_remove(number_remove(remove_user(regex_remove(hashtag_remove(rt_remove(slang_remove(
            re.sub("\n\r\t\xa0", " ", i).strip()))))))))
    clean_text(data["Text"])
```

Gambar 3.6 Source code *tokenization*

Tujuan dari *tokenization* adalah merubah kalimat menjadi potongan kata dan menghapus tanda baca untuk proses analisis teks lebih lanjut.

3. Case Folding

Case folding yaitu proses mengubah semua huruf menjadi huruf kecil di dalam dokumen. Berikut source code dalam *case folding* yang ditampilkan pada Gambar 3.7

```
def lowercase():
    lower_word = data['tweet'].str.lower()
    return lower_word

lower_tweet = lowercase()

print(lower_tweet)
```

Gambar 3.7 Source code *case folding*

Case folding bertujuan mengurangi redundansi data dalam proses pengklasifikasian agar perhitungan bisa optimal.

4. Stopword removal

Stopword removal yaitu melakukan penghapusan kata tidak penting dari sebuah teks, misalnya "di", "oleh", "pada", "sebuah", "karena" dan lain sebagainya, dan menggunakan *library sastrawi* yang dimana fungsi dari *library* tersebut mengenali teks Bahasa Indonesia. Berikut source code dalam proses *stopword removal* yang ditampilkan pada Gambar 3.8

```

from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
stopwords = factory.get_stop_words()
print(stopwords)

```

Gambar 3.8 Source code *stopword removal*

Hasil dari *stopword removal* dari *library* sastrawi ditampilkan pada Tabel 3.4

Tabel 3.4 Daftar kata di dalam *stopword removal*

Daftar Kata library Sastrawi
<pre> ['yang', 'untuk', 'pada', 'ke', 'para', 'namun', 'menurut', 'antara', 'dia', 'dua', 'ia', 'seperti', 'jika', 'jika', 'sehingga', 'kembali', 'dan', 'tidak', 'ini', 'karena', 'kepada', 'oleh', 'saat', 'harus', 'sementara', 'setelah', 'belum', 'kami', 'sekitar', 'bagi', 'serta', 'di', 'dari', 'telah', 'sebagai', 'masih', 'hal', 'ketika', 'adalah', 'itu', 'dalam', 'bisa', 'bahwa', 'atau', 'hanya', 'kita', 'dengan', 'akan', 'juga', 'ada', 'mereka', 'sudah', 'saya', 'terhadap', 'secara', 'agar', 'lain', 'anda', 'begitu', 'mengapa', 'kenapa', 'yaitu', 'yakni', 'daripada', 'itulah', 'lagi', 'maka', 'tentang', 'demi', 'dimana', 'kemana', 'pula', 'sambil', 'sebelum', 'sesudah', 'supaya', 'guna', 'kah', 'pun', 'sampai', 'sedangkan', 'selagi', 'sementara', 'tetapi', 'apakah', 'kecuali', 'sebab', 'selain', 'seolah', 'seraya', 'seterusnya', 'tanpa', 'agak', 'boleh', 'dapat', 'dsb', 'dst', 'dll', 'dahulu', 'dulunya', 'anu', 'demikian', 'tapi', 'ingin', 'juga', 'nggak', 'mari', 'nantinya', 'melainkan', 'oh', 'ok', 'seharusnya', 'sebetulnya', 'setiap', 'setidaknya', 'sesuatu', 'pasti', 'saja', 'toh', 'ya', 'walau', 'tolong', 'tentu', 'amat', 'apalagi', 'bagaimanapun'] </pre>

Dari Tabel 3.4 menunjukkan hasil dari data yang tidak berpengaruh dalam proses analisis. Berikut source code untuk menghilangkan kata-kata yang terdapat pada list *stopword removal* dapat dilihat pada Gambar 3.9

```

def removeStopWords(tweet):
    clean_word_list = [word for word in tweet.split() if word not in stopwords]
    return clean_word_list

stopwords_tweet = lower_tweet.apply(removeStopWords)

print(stopwords_tweet)

```

Gambar 3.9 Source code *stopword*

Source code diatas yaitu sebuah fungsi bernama *removeStopword* digunakan untuk mengecek kata yang terdapat pada *term* sudah sesuai atau belum pada list kata *stopword*. Jika sudah sesuai maka dilakukan penghilangan kata pada *variable clean_word_list*.

5. Stemming

Stemming yaitu proses pemetaan dan penguraian bentuk kata ke bentuk kata dasarnya, Berikut source code yang digunakan dalam proses *stemming* dapat dilihat pada Gambar 3.10

```

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in stopwords_tweet:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ""

print(len(term_dict))
print("-----")

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term,":", term_dict[term])

print(term_dict)
print("-----")

def get_stemmed_term(document):
    return [term_dict[term] for term in document]

stem_tweet = stopwords_tweet.apply(get_stemmed_term)

print(stem_tweet)

```

Gambar 3.10 Source code pada proses *stemming*

Stemming dilakukan dengan mengubah bentuk *dataframe* menjadi bentuk *dictionary* yang nantinya akan menggunakan library *StemmerFactory* untuk mengubah istilah atau kata yang terdapat dalam *term* menjadi kata dasar.

6. Normalisasi

Normalisasi adalah memperbaiki kata-kata yang salah dalam teks berdasarkan korpus yang dibuat, Berikut source code yang digunakan dalam proses normalisasi dapat dilihat pada Gambar 3.11

```

normalizad_word = pd.read_excel("rumahbelajar_kamus.xlsx")

normalizad_word_dict = {}

for index, row in normalizad_word.iterrows():
    if row[0] not in normalizad_word_dict:
        normalizad_word_dict[row[0]] = row[1]

def normalized_term(document):
    return [normalizad_word_dict[term] if term in normalizad_word_dict else term for term in document]

normal_tweet = stem_tweet.apply(normalized_term).str.join("")

print(normal_tweet)

```

Gambar 3.11 Source code normalisasi

normalisasi merupakan proses untuk mengubah kata yang tidak sesuai dengan *list* kata yang sudah dibuat pada file *normalisasi.xlsx* dan kata tersebut menjadi sebuah *dataframe* dengan nama variabel dengan nama *normal_tweet* yang nantinya akan menghasilkan data *tweet* yang sudah di *pre-processing*. Berikut contoh data pada file *normalisasi.xlsx* yang telah dibuat berdasarkan topik pembahasan sebagai dataset memperbaiki kata yang salah dapat dilihat pada Tabel 3.5

Tabel 3.5 Daftar kata pada normalisasi

Kata sebelum	Kata sesudah
Tmn	Teman
Knp	Kenapa
Krn	Karena
Bljr	Belajar
Cpt	Cepat
Udh	Sudah
Drpd	Daripada
Lgsg	Langsung
Karna	Karena

Data yang sudah dilakukan *pre-processing* nantinya akan menghasilkan data *tweet* yang siap digunakan dan datanya lebih terstruktur untuk tahapan selanjutnya, Berikut adalah contoh *tweet pre-processing* dapat dilihat pada Tabel 3.6

Tabel 3.6 Contoh *tweet pre-processing*

No	Hasil Pre-processing
1	langganan ruangguru aja padahal lebih murah
2	belajar ruangguru
3	hai kak maaf paket ruangguru tidak pindah jadi paket internet reguler info lengkap ruangguru cek link tks aisha
4	help zenius sama quipper besok akhir untuk gratis
5	quipper besok habis jam berapa
6	ikut premium zenius
7	zenius aku buka bulan yang lalu padahal udah premium
8	rumah belajar maksimal
9	pandemi covid gunakan aplikasi cakap naik lima kali lipat
10	quipper yang mantep apa sih

3.4.3 Pelabelan manual

Pelabelan manual adalah proses memberikan label terhadap teks atau kalimat suatu dokumen sehingga sifat positif atau negatifnya dapat dianalisis lebih lanjut. Data *tweet* yang sudah dilabeli dengan jumlah data 798 *tweet* dari 24.311 *tweet* dengan masing-masing 399 *tweet* positif dan 399 *tweet* negatif dari data *training*, berikut contoh *tweet* yang sudah dilabeli secara manual dapat dilihat pada Tabel 3.7

Tabel 3.7 Pelabelan manual

No	Tweet	label	kelas
1	ayo kak langganan ruang guru nya	positif	1
2	belajar ruang guru bagus kok	positif	1
3	ayo buka ruangguru	positif	1

4	semangatt coba belajar pake quipper aja enak banget buat belajar ekonomi	positif	1
5	coba quipper enak banget tutor nyaaaa	positif	1
6	anjir notice ngakak bgttt	negatif	0
7	lu mana anjir kayak ruang guru	negatif	0
8	capek buka tutup app nya	negatif	0
9	ko kalau ngerjain soal ruangguru kayak aja kalau yang kek susah banget	negatif	0
10	aku bingung banget mau belajar darimana lama rumah malas malesan terus	negatif	0

3.4.4 Data Training

Pada proses data *training* diawali dengan fitur ekstraksi pada data teks menggunakan TF-IDF dan pada tahap ini menggunakan metode *Support Vector Machine*, kemudian dilakukan proses *training* untuk membuat model klasifikasi dalam bentuk format *pickle* yang dimana dapat digunakan untuk melakukan *klasifikasi* sentimen secara otomatis. Berikut contoh perhitungan TF-IDF secara manual dengan Microsoft Office Excel dapat dilihat pada Tabel 3.8

Tabel 3.8 Dokumen TF-IDF

Dokumen (d)	Kalimat
d1	enak quipper dulu bang terus lanjut zenius terus buat nginget nginget baca rangkum ruangguru
d2	pandemi covid gunakan aplikasi cakap naik lima kali lipat
d3	quipper bikin kompetisi pekan didik nasional
d4	jujur zenius yang baru gaenak banget banyak eror loadingnya jadi lama banget kalau to capek skroling teks

Pada Tabel 3.6 terdapat beberapa kalimat yang dapat digunakan untuk perhitungan manual TF-IDF dengan 4 dokumen yaitu d1, d2, d3 dan d4. Dalam melakukan perhitungan *Term Frequency* (TF) ini menggunakan beberapa komponen yaitu *term* atau kata, dan *d* merupakan jumlah data yang akan digunakan terdiri dari d1, d2, d3 dan d4 dan *df* untuk menghitung jumlah term

atau kata yang muncul pada setiap dokumen. Contoh perhitungan TF dapat dilihat pada Tabel 3.9

Tabel 3.9 Perhitungan TF

Term (kata)	d1	d2	d3	d4	df
enak	1				1
quipper	1		1		2
dulu	1				1
bang	1				1
terus	1				1
lanjut	1				1
buat	1				1
nginget	2				2
baca	1				1
rangkum	1				1
ruangguru	1				1
pandemi		1			1
covid		1			1
gunakan		1			1
aplikasi		1			1
cakap		1			1
naik		1			1
lima		1			1
kali		1			1
lipat		1			1
bikin			1		1
kompetisi			1		1
pekan			1		1
didik			1		1
nasional			1		1
jujur				1	1
zenius				1	1

yang				1	1
baru				1	1
gaenak				1	1
banget				2	2
banyak				1	1
error				1	1
loadingnya				1	1
jadi				1	1
lama				1	1
kalau				1	1
to				1	1
capek				1	1
skroling				1	1
teks				1	1

Pada Tabel 3.9 menjelaskan kemunculan *term* atau kata pada kalimat setiap dokumen. Perhitungan *Invers Document Frequency* (IDF) menggunakan beberapa komponen seperti *term* atau kata, *df* dan *idf* yang berhubungan antara ketersediaan suatu *term* di semua dokumen, dihitung dengan *N* atau jumlah dokumen. Berikut contoh perhitungan IDF pada Tabel 3.10

Tabel 3.10 Perhitungan IDF

term (kata)	df	idf	idf (N=4)	idf (N=800)
enak	1	1	0.60206	2,90309
quipper	2	0.5	0.30103	2,60206
dulu	1	1	0.60206	2,90309
bang	1	1	0.60206	2,90309
terus	1	1	0.60206	2,90309
lanjut	1	1	0.60206	2,90309
buat	1	1	0.60206	2,90309
nginget	2	0,5	0.30103	2,60206
baca	1	1	0.60206	2,90309

rangkum	1	1	0.60206	2,90309
ruangguru	1	1	0.60206	2,90309
pandemi	1	1	0.60206	2,90309
covid	1	1	0.60206	2,90309
gunakan	1	1	0.60206	2,90309
aplikasi	1	1	0.60206	2,90309
cakap	1	1	0.60206	2,90309
naik	1	1	0.60206	2,90309
lima	1	1	0.60206	2,90309
kali	1	1	0.60206	2,90309
lipat	1	1	0.60206	2,90309
bikin	1	1	0.60206	2,90309
kompetisi	1	1	0.60206	2,90309
pekan	1	1	0.60206	2,90309
didik	1	1	0.60206	2,90309
nasional	1	1	0.60206	2,90309
jujur	1	1	0.60206	2,90309
zenius	1	1	0.60206	2,90309
yang	1	1	0.60206	2,90309
baru	1	1	0.60206	2,90309
gaenak	1	1	0.60206	2,90309
banget	2	0,5	0.30103	2,60206
banyak	1	1	0.60206	2,90309
error	1	1	0.60206	2,90309
loadingnya	1	1	0.60206	2,90309
jadi	1	1	0.60206	2,90309
lama	1	1	0.60206	2,90309
kalau	1	1	0.60206	2,90309
to	1	1	0.60206	2,90309
capek	1	1	0.60206	2,90309
skroling	1	1	0.60206	2,90309

teks	1	1	0.60206	2,90309
------	---	---	---------	---------

Pada Tabel 3.10 menunjukkan mengenai perhitungan idf secara manual dengan rumus persamaan (1). Berikut penjelasan mengenai perhitungan TF-IDF pada Tabel 3.11

Tabel 3.11 Perhitungan TF-IDF

term (kata)	d1	d2	d3	d4
enak	0.60206			
quipper	0.30103		0.30103	
dulu	0.60206			
bang	0.60206			
terus	0.60206			
lanjut	0.60206			
buat	0.60206			
nginget	0.60206			
baca	0.60206			
rangkum	0.60206			
ruangguru	0.60206			
pandemi		0.60206		
covid		0.60206		
gunakan		0.60206		
aplikasi		0.60206		
cakap		0.60206		
naik		0.60206		
lima		0.60206		
kali		0.60206		
lipat		0.60206		
bikin			0.60206	
kompetisi			0.60206	
pekan			0.60206	
didik			0.60206	

nasional			0.60206	
jujur				0.60206
zenius				0.60206
yang				0.60206
baru				0.60206
gaenak				0.60206
banget				0.60206
banyak				0.60206
error				0.60206
loadingnya				0.60206
jadi				0.60206
lama				0.60206
kalau				0.60206
to				0.60206
capek				0.60206
skroling				0.60206
teks				0.60206

Pada Tabel 3.11 menjelaskan perhitungan TF-IDF secara manual dengan menggunakan Microsoft Office Excel dari hasil perkalian tf dan idf.

3.4.5 Testing

Pada tahap *testing* adalah sebuah proses menentukan tingkat akurasi dari pemodelan yang dibuat pada tahap pelatihan, digunakan untuk memprediksi label atau kelas dari data uji yang tersedia. Maka ditampilkan hasil dari penggabungan pelabelan secara manual (*actual*) dan juga svm (*predicted*). Berikut hasilnya dapat dilihat pada Gambar 3.12

no	tweet	label	predict	keterangan	
0	1	aaaa aku gayakin aku ngajarin aku cuma anak su...	P	P	TP
1	2	aamiin jelek hyung	N	N	TN
2	3	adek gua minta ajarin kimia gua suka banget ki...	P	N	FP
3	4	akhir gw mutusin kek juni mau rehat dulu tidak...	N	N	TN
4	5	akhir tidak sendiri truss punya masih kyk gt a...	N	N	TN
...
395	396	yon tidak mau lamar kerja jadi guru quipper	N	P	FN
396	397	zeennnn jadi kapan sih intensif nya hari lg di...	N	N	TN
397	398	zen video bahas simak tidak lengkap matdas	N	N	TN
398	399	zeni apa kamu tidak mau beri yang masih us lan...	N	N	TN
399	400	zenius edukun sama pahamify aku ngerjain soal ...	P	P	TP

400 rows × 5 columns

Gambar 3.12 Hasil pelabelan manual

Kemudian mencari nilai pemodelan data pada Gambar 3.12 meliputi *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *True Negative* (TN). Agar bisa dilakukan penghitungan nilai *testing* dari *accuracy*, *precision*, *recall* dan *f-measure*. Berikut *source code* pemodelannya seperti dibawah ini pada Gambar 3.13 sampai Gambar 3.16

```
: accuracy = (TP+TN)/(TP+TN+FP+FN)
print('Accuracy =', accuracy)
```

Accuracy = 0.7275

Gambar 3.13 Hasil *Accuracy*

```
precision = (TP) / (TP+FP)
print('Precision =', precision)
```

Precision = 0.715

Gambar 3.14 Hasil *Precision*

```
recall = (TP) / (TP + FN)
print('Recall =', recall)

Recall = 0.7333333333333333
```

Gambar 3.15 Hasil *Recall*

```
F1_Score = 2 * (recall*precision) / (recall + precision)
print('F1Score =', F1_Score)

F1Score = 0.7240506329113924
```

Gambar 3.16 Hasil *F1Score*

3.4.6 Hasil Klasifikasi

Setelah mendapatkan nilai akurasi yang baik dari proses *training* dan *testing* dari pemodelan klasifikasi maka dari itu dilakukannya tahap klasifikasi untuk data keseluruhan agar mendapatkan hasil sentimen positif dan negatif yang telah diuji dari tahapan *training* dan *testing*. Berikut hasil dari klasifikasi keseluruhan data dapat dilihat pada Gambar 3.17

	tweet	class
0	aplikasi buat belajar bebs aku aplikasi etapi ...	P
1	akses pihak tiga luar sistem cakap dapat aplik...	P
2	hai kak ifa maaf sudah bikin ga nyaman kena ku...	P
3	punya nervous nak kena cakap menteri memang ti...	P
4	dirumahterusbelajar bahasa inggris pakai aplik...	P
...
23900	umur belajar pantas	N
23901	nonton materi tps jamin	P
23902	paham konsep aku sini belajar zenius dulu jadi...	N
23903	terima kasih banyakk zeniusssss	P
23904	buku wangsit education buku nf tambah zenius	P

23905 rows × 2 columns

Gambar 3.17 Hasil sentimen data keseluruhan