

BAB 3

METODE PENELITIAN

Penelitian ini adalah penelitian rancang-bangun. Penelitian dimulai dari latar belakang permasalahan, menggambarkan proses, mencari akar permasalahan, dan kemudian menganalisis pengalaman pengguna yang diambil dari Twitter dengan harapan dapat mengurangi masalah yang ada. Berikut adalah bahan, alat, metode pengembangan sistem, dan tahapan penelitian yang dibutuhkan pada penelitian ini.

3.1 BAHAN DAN ALAT PENELITIAN

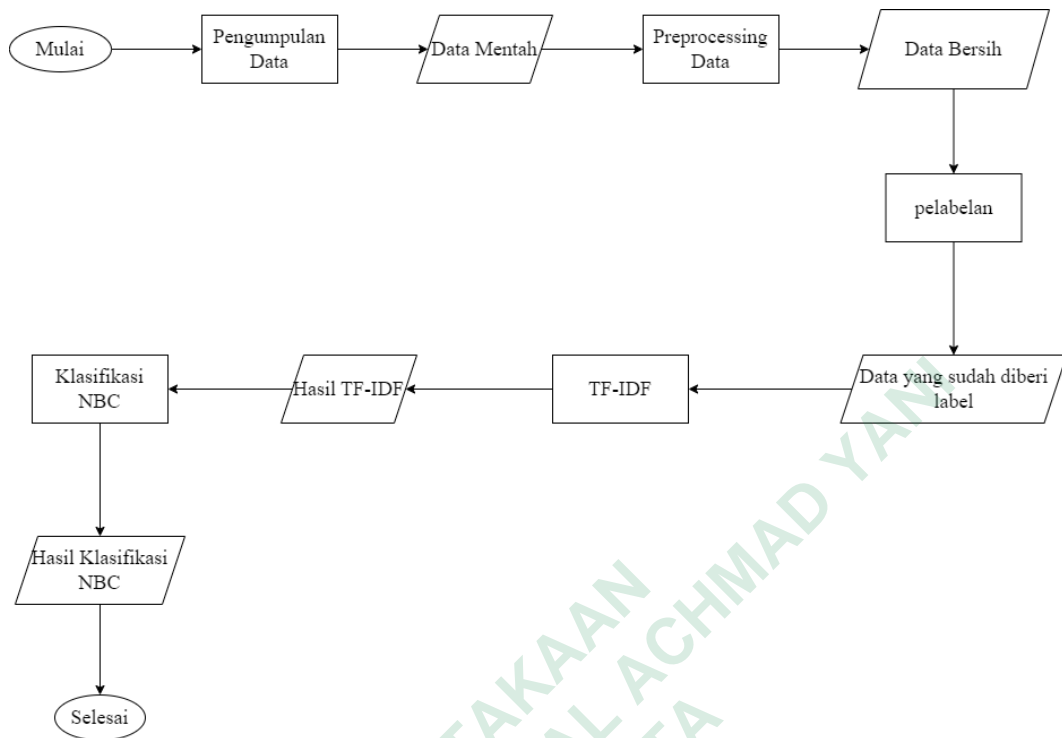
Bahan penelitian yang diperlukan untuk penelitian ini adalah opini-opini masyarakat terkait topik transportasi umum dengan *keyword* Trans Jogja, KRL Solo-Jogja dan KA Bandara YIA sebanyak 3267 data. Selain itu ada beberapa bahan yang menjadi referensi, seperti jurnal penelitian terdahulu, buku dan skripsi yang relevan dengan pembahasan yang dibutuhkan.

Alat yang dipakai dalam penelitian ini adalah komputer dengan spesifikasi yang mendukung untuk pengembangan perangkat lunak dan menjalankan sistem operasi, serta koneksitas Internet. Dengan program aplikasi dan sistem operasi yang digunakan dalam pengembangan aplikasi ini adalah sebagai berikut:

1. Sistem Operasi : Windows 10
2. Prosesor : Intel® Core™ i3-6006U CPU @ 2.00GHz
1.99GHz
3. Bahasa Pemrograman : Python 3.11
4. Text Editor : Visual Studio Code & Google Colab

3.2 JALAN PENELITIAN

Jalan penelitian yang akan digunakan untuk Analisis Sentimen terhadap Layanan Jasa Transportasi Umum di Daerah Istimewa Yogyakarta pada Media Sosial Twitter dengan Metode Naïve Bayes dapat dilihat pada Gambar 3.2.1.



Gambar 3.1 Alur Penelitian

3.3 TAHAP PENGUMPULAN DATA

Langkah pertama dalam riset ini adalah fase pengumpulan data dari tweet menggunakan *tools* tweet harvest dan proses ini dilakukan dengan menggunakan Google Colab. Dataset yang diperoleh dari proses pengambilan data ini memiliki batasan dalam hal jumlah data yang dapat diambil. Oleh karena itu, pengambilan data dilakukan secara berkala. Prosedur pengambilan data ini memiliki persyaratan, yaitu perangkat harus terhubung dengan internet dan dilengkapi dengan API Twitter yang memerlukan token autentikasi untuk menghubungkan ke program tertentu. Proses penambahan data ini terbatas pada kata kunci tertentu, yaitu Trans Jogja, KRL Jogja-Solo, dan KA Bandara YIA. Setelah kata kunci ditetapkan, langkah selanjutnya yaitu menentukan jumlah data yang akan diambil dalam riset ini. Setelah proses selesai dilakukan data yang diperoleh sebanyak 12.000 data, data mentah yang diperoleh akan disimpan dalam format file CSV agar dapat digunakan pada tahap berikutnya dalam riset ini. Hasil dari pengumpulan data bisa dilihat pada Tabel 3.1 berikut ini.

Tabel 3.1 Hasil Pengumpulan Data

Full_text	Username
kereta bandara YIA kenapa belum ada jadwalnya min :(H-2 loh aku nih huhu @KAI121	chelsmwd
kereta bandara dari YIA ke kota tuh masih ada ga sih ? @KAI121	yelyeahwilliams
@KAIBandara Info dong, pemesanan tiket kereta bandara YIA ke Yogyakarta tanggal 10 agustus, kenapa tidak bisa ya?mohon infonya, tks.	tomijo_geol
naik kereta bandara YIA dari stasiun tugu, lengang banget kakinya bisa selonjor dengan harga 20rb waktu perjalanan 40 menit https://t.co/gd3SkL2Ffc	maiaaaddict

3.4 TAHAP PREPROCESSING

Langkah yang mengikuti *scrapping* data tweet adalah tahap *preprocessing*. Pada tahap ini, data tweet yang telah didapat akan mengalami proses pembersihan dari simbol, angka, tagar (*hashtag*), emoji dan tautan. Selain itu, kata-kata yang dianggap tidak memiliki signifikansi akan dihapus, dan tweet akan melalui proses normalisasi di mana kata-kata yang tidak standar akan diubah menjadi bentuk standar. Berikut ini kode pada proses preprocessing dapat dilihat pada Gambar 3.2 dan Gambar 3.3

```
def allowed_files(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSION

def prepropeocessing_twitter():
    global normalizad_word_dict
    normalizad_word = pd.read_excel("templates/assets/normalisasi.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]

    # Membuat File CSV
    file = open("templates/assets/files/Data Preprocessing Agnes.csv", 'w', newline='', encoding='utf-8')
    writer = csv.writer(file)

    hasil_preprocessing.clear()

    with open("templates/assets/files/Data Scrapping Agnes.csv", "r", encoding='utf-8') as csvfile:
        readCSV = csv.reader(csvfile, delimiter=',')
        hasil_labelling.clear()
        for row in readCSV:
            # proses clean
            clean = ' '.join(re.sub("([A-Za-z0-9+])|([^\0-9A-Za-z \t])|(\w+:\w+\/\w+\/\w+)", "", row[2]).split())
            clean = re.sub("\d+", "", clean)
            clean = re.sub(r"[a-zA-Z]\b", "", clean)

            # proses casefolding
            casefold = clean.casefold()

            # proses normalize
            normalisasi = proses_normalisasi(casefold)

            # proses tokenize
            tokenizing = nltk.tokenize.word_tokenize(normalisasi)

            # proses stop removal
            # mengambil data stop word dari library
            stop_factory = StopWordRemoverFactory().get_stop_words()
```

Gambar 3 2 Kode Program Preprocessing

```

92 # menambahkan stopwords sendiri
93 more_stop_word = [ "apa", "yg", 'yg', 'dg', 'rt', 'dgn', 'ny', 'd', 'klo',
94                  'kalo', 'amp', 'bian', 'bikin', 'bilang',
95                  'gak', 'ga', 'knn', 'nya', 'nih', 'sih',
96                  'si', 'tau', 'tdk', 'tuh', 'utk', 'ya',
97                  'jd', 'jgn', 'sdh', 'aja', 'n', 't', 'dm',
98                  'nyg', 'hehe', 'pen', 'u', 'nan', 'loh', 'rt',
99                  '&amp;', 'yah', 'hallo', 'halo', 'hello', 'bgt', 'td',
100                 'no', 'yaa', 'ae', 'kali', 'segera', 'rd', 'kak', 'gmn', 'min']
101 # menggabungkan stopwords library + milik sendiri
102 data = stop_factory + more_stop_word
103
104 dictionary = ArrayDictionary(data)
105 str = StopWordRemover(dictionary)
106 stop_wr = nltk.tokenize.word_tokenize(str.remove(casefold))
107
108 # proses stemming
109 kalimat = ' '.join(stop_wr)
110 factory = StemmerFactory()
111 # memanggil fungsi stemming
112 stemmer = factory.create_stemmer()
113 stemming = stemmer.stem(kalimat)
114
115 tweets =[row[0], row[1], row[2], clean, casefold, normalisasi, tokenizing, stop_wr, stemming]
116 hasil_preprocessing.append(tweets)
117
118 writer.writerow(tweets)
119 flash('Preprocessing Berhasil', 'preprocessing_data')
120
121
122 def normalized_term(mosok):
123     return [normalized_word_dict[term] if term in normalized_word_dict else term for term in mosok]
124
125 def proses_normalisasi(data):
126     tokens = nltk.tokenize.word_tokenize(data)
127     hasil = normalized_term(tokens)
128     kalimat = ' '.join(hasil)
129     return kalimat

```

Gambar 3.3 Kode Program Preprocessing

1. *Cleaning*, Langkah pembersihan dilakukan untuk mengeliminasi tautan , tanda baca, *mention*, tagar, dan angka guna memastikan bahwa berada dalam kondisi yang bersih. Sistem akan mengidentifikasi karakter khusus yang telah ditentukan sebelumnya, dan karakter tersebut akan dibersihkan dari setiap tweet.
2. *Case Folding*, Proses ini dilaksanakan agar semua huruf menjadi huruf kecil (*lower case*) dan format huruf menjadi seragam. Pada tahap ini, sistem memeriksa setiap tweet dari awal hingga akhir, jika ditemukan huruf kapital, maka huruf tersebut akan diubah menjadi huruf kecil.
3. *Normalisasi*, Langkah normalisasi ditujukan untuk mengubah kata-kata singkatan, kata-kata slang, serta kata-kata non-baku menjadi bentuk yang standar. Normalisasi dilakukan dengan memanfaatkan kamus istilah non-baku.csv di dalam sistem. Pada tahap ini, sistem akan mengamati seluruh tweet, jika terdapat suatu kata yang tidak sesuai dengan KBBI, kata tersebut

akan diganti menjadi bentuk yang sesuai, sebagai contoh, mengubah "dg" menjadi "dengan," "bgt" menjadi "banget," dan "tdk" menjadi "tidak."

4. *Tokenizing*, Langkah tokenisasi merupakan langkah awal dalam Pemrosesan Bahasa Alami (NLP). Proses tokenisasi dilakukan untuk memecah string menjadi unit-token dengan menggunakan pustaka NTLK. Tahap ini, sistem akan melacak semua dan mengubah string menjadi token di dalam setiap tweet, yang akan diapit oleh tanda kutip dan dipisahkan oleh koma.
5. *Stopword Removal*, Proses penghapusan kata hubung bertujuan untuk menghilangkan kata-kata yang tidak memberikan makna signifikan dalam sebuah tweet. Kata-kata ini dihapus karena tidak mewakili esensi dari tweet tersebut. Tahap ini, sistem akan menelusuri semua tweet, dan jika ditemukan kata hubung yang tidak relevan, kata tersebut akan dihapus. Proses ini menggunakan kamus kata-kata hubung dari pustaka, dan bisa juga ditambahkan dengan daftar kata-kata hubung khusus.
6. *Stemming*, Langkah *stemming* adalah proses untuk menghilangkan awalan atau akhiran dari suatu kata dalam tweet agar hanya tersisa bentuk dasar kata, dan ini dilakukan menggunakan pustaka sastra yang sesuai dengan bahasa Indonesia. Pada tahap ini, sistem akan mengikuti semua tweet dari awal hingga akhir, dan jika terdapat kata-kata yang mengandung imbuhan, kata-kata tersebut akan diubah menjadi bentuk dasar.

Langkah selanjutnya setelah mengumpulkan adalah melakukan *preprocessing* pada teks tweet. Tahap ini, data tweet yang telah didapat akan dibersihkan dari tagar, angka, tautan, dan simbol. Lalu kata-kata yang tidak memiliki arti dalam tweet akan disingkirkan, dan tweet akan melalui proses pengubahan kata dari yang tidak baku menjadi kata baku. Hasil preprocessing data bisa dilihat pada tabel 3s.2 berikut ini.

Tabel 3.2 Hasil Preprocessing Data

tweet	cleaning	casefolding	normalisasi	tokenizing	stopword removal	stemming
kereta bandara YIA kenapa belum ada jadwalnya min :(H-2 loh aku nih huhu @KAI121	kereta bandara YIA kenapa belum ada jadwalnya min loh aku nih huhu	kereta bandara yia kenapa belum ada jadwalnya min loh aku nih huhu	kereta bandara yia kenapa belum ada jadwalnya min loh aku nih huhu	['kereta', 'bandara', 'yia', 'kenapa', 'belum', 'ada', 'jadwalnya', 'min', 'loh', 'aku', 'nih', 'huhu']	['kereta', 'bandara', 'yia', 'belum', 'jadwalnya', 'aku', 'huhu']	kereta bandara yia belum jadwal aku huhu
kereta bandara dari YIA ke kota tuh masih ada ga sih ? @KAI121,	kereta bandara dari YIA ke kota tuh masih ada ga sih	kereta bandara dari yia ke kota tuh masih ada ga si	kereta bandara dari yia ke kota tuh masih ada enggak sih	['kereta', 'bandara', 'dari', 'yia', 'ke', 'kota', 'tuh', 'masih', 'ada', 'enggak', 'sih']	'kereta', 'bandara', 'yia', 'kota', 'masih', 'ga	kereta bandara yia kota masih ga

3.5 TAHAP PELABELAN ATAU LABELING

Proses pemberian label adalah langkah untuk mengidentifikasi sentimen dari sebuah tweet, yang kemudian dapat diberi label positif, negatif, atau netral. Dalam tahap pemberian label ini, digunakan pustaka atau *library* TextBlob. TextBlob adalah salah satu pustaka yang telah disediakan oleh bahasa pemrograman Python untuk melakukan pemrosesan di bidang Pemrosesan Bahasa Alami (Natural Language Processing/NLP). Namun, perlu dicatat bahwa saat ini pustaka TextBlob belum mendukung bahasa Indonesia secara langsung. Oleh karena itu, dataset yang telah melalui tahap *stemming* dalam bahasa Indonesia diterjemahkan ke dalam bahasa Inggris agar dapat diproses menggunakan TextBlob. Proses pemberian label dilakukan dengan memanfaatkan konsep polaritas. Nilai polaritas digunakan untuk menentukan label sebuah tweet. Tweet dengan nilai polaritas 1 akan diberi label positif, nilai polaritas 0 akan diberi label netral, sementara nilai polaritas -1 akan diberi label negatif.

Hasil dari proses pelabelan dapat dilihat pada Tabel 3.3.

Tabel 3.3 Hasil Labelling

tweet	label
kereta bandara yia belum jadwal aku huhu	Netral
,kereta bandara yia kota masih ga	Netral

Setelah pelabelan dilakukan pada 3267 data lalu dapat diketahui jumlah data positif, negatif, dan netral. Adapun jumlah tiap label dilihat pada Table 3.4.

Tabel 3.4 Jumlah Data Perlabel

Label	Jumlah
Positif	1452
Negatif	517
Netral	1298

Berikut ini kode pada proses preprocessing dapat dilihat pada Gambar 3.4

```

1 def labelling_process():
2     # Membuat File CSV
3     file = open('templates/assets/files/Data Labelling Agnes.csv', 'w', newline='', encoding='utf-8')
4     writer = csv.writer(file)
5     translator = Translator()
6
7     with open("templates/assets/files/Data Preprocessing Agnes.csv", "r",encoding='utf-8') as csvfile:
8         readCSV = csv.reader(csvfile, delimiter=',')
9         hasil_labelling.clear()
10        for row in readCSV:
11            tweet = {}
12            try:
13                value = translator.translate(row[8], dest='en')
14                terjemahan = value.text
15                data_label = TextBlob(terjemahan)
16
17                if data_label.sentiment.polarity > 0.0 :
18                    tweet['sentiment'] = "Positif"
19                elif data_label.sentiment.polarity == 0.0 :
20                    tweet['sentiment'] = "Netral"
21                else :
22                    tweet['sentiment'] = "Negatif"
23                labelling = tweet['sentiment']
24                tweets =[row[1], row[8], labelling]
25                hasil_labelling.append(tweets)
26
27                print(tweets)
28
29                writer.writerow(tweets)
30                flash('Labelling Berhasil', 'labelling_data')
31            except:
32                print("Terjadi kesalahan", flush=True)

```

Gambar 3.4 Kode Program Proses Labelling

3.6 TAHAP KLASIFIKASI

Dataset tweet yang telah melalui langkah-langkah pra-pemrosesan dan pemberian label menggunakan TextBlob akan diarahkan ke fase klasifikasi menggunakan metode Naïve Bayes. Berikut adalah contoh perhitungan klasifikasi kelas sentiment dengan menggunakan metode naïve bayes dengan data yang terdiri dari 13 data latih dan 1 data uji. Terdapat 3 data dengan kelas positif, 5 data dengan kelas negative dan 2 data dengan kelas netral. Pertama menghitung terlebih dahulu probabilitas setiap kelas. Adapun kumpulan data dapat dilihat pada table 3.5.

Tabel 3. 5 Contoh Kasus

No	Data	Kelas
D1	ka bandara yia kurang dingin ac	Negatif
D2	krl jogja solo lambat lama	Negatif
D3	kereta bandara yia viewnya bagus	Positif
D4	cara naik trans jogja gimana	Netral
D5	krl jogja solo murah bang	Positif
D6	ini lambat krl jogja solo	Negatif
D7	krl jogja solo bagus banget	Positif
D8	mau pesan tiket kereta bandara tugu yia oktober kok ada jadwal apa operasi gimana	Negatif
D9	krl solo jogja banyak orang daerah gabisa henti ngobrol mana bawa anak berisik gue capek mau pulang tenang jadi gabisa	Negatif
D10	ka bandara yia emang favorit banget bener mangkas waktu jalan bayangin menit jam orang pada pilih paling cepat kan	Netral
D11	mending naik trans jogja murah	?

1. Menentukan probabilitas kelas

- a. Positif $\rightarrow 3/10 = 0,3$
- b. Negatif $\rightarrow 5/10 = 0,5$
- c. Netral $\rightarrow 2/10 = 0,2$

2. Menentukan probabilitas kategori

a. Kelas Positif

$$P(\text{Mending} | \text{Positif}) = (0+1)/15+15 = 0,033$$

$$P(\text{Naik} | \text{Positif}) = (0+1)/15+15 = 0,033$$

$$P(\text{Trans} | \text{Positif}) = (0+1)/15+15 = 0,033$$

$$P(\text{Jogja} \mid \text{Positif}) = (0+1)/15+15 = 0,033$$

$$P(\text{Murah} \mid \text{Positif}) = (1+1)/15+15 = 0,066$$

b. Kelas Negatif

$$P(\text{Mending} \mid \text{Negatif}) = (0+1)/50+15 = 0,015$$

$$P(\text{Naik} \mid \text{Negatif}) = (0+1)/50+15 = 0,015$$

$$P(\text{Trans} \mid \text{Negatif}) = (0+1)/50+15 = 0,015$$

$$P(\text{Jogja} \mid \text{Negatif}) = (3+1)/50+15 = 0,061$$

$$P(\text{Murah} \mid \text{Negatif}) = (0+1)/50+15 = 0,015$$

c. Kelas Netral

$$P(\text{Mending} \mid \text{Netral}) = (0+1)/24+15 = 0,025$$

$$P(\text{Naik} \mid \text{Netral}) = (1+1)/24+15 = 0,051$$

$$P(\text{Trans} \mid \text{Netral}) = (1+1)/24+15 = 0,051$$

$$P(\text{Jogja} \mid \text{Netral}) = (1+1)/24+15 = 0,051$$

$$P(\text{Murah} \mid \text{Netral}) = (0+1)/24+15 = 0,025$$

Setelah nilai probabilitas setiap kelas dihitung, selanjutnya dilakukan penentuan klasifikasi suatu tweet. Didapatkan hasil sebagai berikut.

a. Kelas Positif

$$P(V1 \mid C) = 0,033 * 0,033 * 0,033 * 0,033 * 0,066 = 7.827$$

b. Kelas Negatif

$$P(V1 \mid C) = 0,015 * 0,015 * 0,015 * 0,061 * 0,015 = 3.088$$

c. Kelas Netral

$$P(V1 \mid C) = 0,025 * 0,051 * 0,051 * 0,051 * 0,025 = 8.290$$

Dari perhitungan diatas , maka di peroleh hasil untuk kelas positif sebesar 7.827, kelas negatif sebesar 3.088 dan kelas netral sebesar 8.290. Dari hasil tersebut, nilai probabilitas pada kelas Netral lebih besar dibandingkan kelas positif dan kelas negatif maka data uji pada D11 termasuk kedalam kelas netral.

Dalam tahap klasifikasi pada sistem ini, sistem memiliki dua opsi pendekatan. Pertama, pendekatan otomatis di mana pengguna dapat melanjutkan proses dari tahap pemberian label dengan mengklik tombol "Lanjutkan," sehingga data akan secara langsung diteruskan ke tahap klasifikasi. Kedua, pengguna juga

dapat menggunakan pendekatan dengan mengunggah berkas labelling.csv, yang kemudian akan digunakan untuk proses klasifikasi. Kode program dari proses klasifikasi dapat dilihat pada Gambar 3.5, Gambar 3.6 dan Gambar 3.7.

```
def klasifikasi_data():
    global df
    global df2
    global akurasi

    # membaca csv
    data = pd.read_csv("templates/assets/files/Data Labelling Agnes.csv")
    tweet = data.iloc[:, 1]
    y = data.iloc[:, 2]

    #split data training dan testing
    x_train, x_test, y_train, y_test = train_test_split(tweet,y, test_size=0.2, random_state=42)
    # tfidf
    vectorizer = TfidfVectorizer(max_features=5000)
    vectorizer.fit(tweet)
    # tfidf = vectorizer.fit_transform(X_train)
    names = vectorizer.get_feature_names_out()
    Train_X_Tfidf= vectorizer.transform(x_train)
    Test_X_Tfidf= vectorizer.transform(x_test)

    # naive bayes
    clf = MultinomialNB()
    clf.fit(Train_X_Tfidf, y_train)

    # menyimpan tfidf
    df_train_tfidf = pd.DataFrame(data=csr_matrix.todense(Train_X_Tfidf))
    df_train_tfidf.columns = names
    df_train_tfidf.index = y_train
    df_train_tfidf.to_csv('templates/assets/files/TFIDF Training.csv')

    df_test_tfidf = pd.DataFrame(data=csr_matrix.todense(Test_X_Tfidf))
    df_test_tfidf.columns = names
    df_test_tfidf.index = y_test
    df_test_tfidf.to_csv('templates/assets/files/TFIDF Testing.csv')
```

Gambar 3.5 Kode Program Proses Klasifikasi

```
pickle.dump(vectorizer, open('templates/assets/files/countvec.pkl', 'wb'))
pickle.dump(Train_X_Tfidf, open('templates/assets/files/tfidf.pkl', 'wb'))
pickle.dump(clf, open('templates/assets/files/model.pkl', 'wb'))

predict = clf.predict(Test_X_Tfidf)

report = classification_report(y_test, predict, output_dict=True)
# simpan ke csv
clsf_report = pd.DataFrame(report).transpose()
clsf_report.to_csv('templates/assets/files/Data Hasil Klasifikasi.csv', index= True)

unique_label = np.unique([y_test, predict])
cmtx = pd.DataFrame(
    confusion_matrix(y_test, predict, labels=unique_label),
    index=['{:}'.format(x) for x in unique_label],
    columns=['{:}'.format(x) for x in unique_label]
)
```

Gambar 3.6 Kode Program Proses Klasifikasi

```

cmtx.to_csv('templates/assets/files/Data Confusion Matrix.csv', index= True)

df = pd.read_csv('templates/assets/files/Data Confusion Matrix.csv', sep=",")
df.rename( columns={'Unnamed: 0':''}, inplace=True )

df2 = pd.read_csv('templates/assets/files/Data Hasil Klasifikasi.csv', sep=",")
df2.rename( columns={'Unnamed: 0':''}, inplace=True )

akurasi = round(accuracy_score(y_test, predict) * 100, 2)

kalimat = ""

```

Gambar 3.7 Kode Program Proses Klasifikasi

3.7 VISUALISASI

Hasil akhir dari proses klasifikasi akan disajikan dalam bentuk grafik batang, grafik lingkaran, serta visualisasi menggunakan wordcloud. Dengan kode program proses visualisasi dapat dilihat pada Gambar 3.8 dan Gambar 3.9

```

for i in tweet.tolist():
    s = "".join(i)
    kalimat += s

urlretrieve("https://firebasestorage.googleapis.com/v0/b/sentimen-97d49.appspot.com/o/Circle.png?alt=media&token=...", "love.png")
mask = np.array(Image.open("love.png"))
wordcloud = WordCloud(width=1600, height=800, max_font_size=200, background_color='white', mask = mask)
wordcloud.generate(kalimat)

plt.figure(figsize=(12,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.savefig('templates/assets/files/wordcloud.png')

# diagram
numbers_list = y_test.tolist()
counter = dict((i, numbers_list.count(i)) for i in numbers_list)
isPositive = 'Positif' in counter.keys()
isNegative = 'Negatif' in counter.keys()
isNeutral = 'Netral' in counter.keys()

positif = counter["Positif"] if isPositive == True else 0
negatif = counter["Negatif"] if isNegative == True else 0
netral = counter["Netral"] if isNeutral == True else 0

sizes = [positif, netral, negatif]
labels = ['Positif', 'Netral', 'Negatif']
plt.pie(sizes, labels=labels, autopct='%1.0f%%', shadow=True, textprops={'fontsize': 20})
plt.savefig('templates/assets/files/diagram.png')

# diagram batang
# creating the bar plot

plt.figure()

plt.hist(numbers_list)

plt.xlabel("Tweet tentang Transportasi Umum DIY")

```

Gambar 3.8 Kode Program Proses Visualisasi

```
hasil_model_predict = []
def model_predict():
    global df
    global df2
    global akurasi
    # membca csv
    data = pd.read_csv("templates/assets/files/Data Labelling Model Predict Agnes.csv")
    tweet = data.iloc[:, 1]
    y = data.iloc[:, 2]

    # Vectorize text reviews to numbers
    # tfidf = joblib.load('templates/assets/files/tfidf.pkl')
    # nb = joblib.load('templates/assets/files/model.pkl')
    # vec = joblib.load('templates/assets/files/countvec.pkl')
```

Gambar 3.9 Kode Program Proses Visualisasi

PERPUSTAKAAN
UNIVERSITAS JENDERAL ACHMAD YANI
YOGYAKARTA