

**LAPORAN AKHIR**  
**PROJECT-BASED LEARNING**  
**MATA KULIAH ALGORITMA DAN PEMROGRAMAN**  
**LANJUT**  
**TAHAP 1: RANCANGAN *SCRIPT* PEMROGRAMAN FUNGSIONAL**  
**DAN/ATAU OOP PADA STUDI KASUS**  
**NATURAL LANGUAGE PROCESSING**  
**KELAS A**



**“ANALISIS SENTIMEN *CYBERBULLYING* PADA MEDIA SOSIAL  
MENGUNAKAN METODE *SUPPORT VECTOR MACHINE*”**

**DISUSUN OLEH KELOMPOK “VI” :**

- |                            |                           |
|----------------------------|---------------------------|
| 1. ALVIN RYAN DANA         | ( 21083010035 ) - KETUA   |
| 2. AMANDA AYU DEWI LESTARI | ( 21083010008 ) - ANGGOTA |
| 3. ADELIA AZIZATUL HAQ     | ( 21083010009 ) - ANGGOTA |
| 4. MOHAMMAD NIZAR RISWANDA | ( 21083010015 ) - ANGGOTA |
| 5. CHELSEA AYU ADHIGIADANY | ( 21083010028 ) - ANGGOTA |

**DOSEN PENGAMPU:**

TRESNA MAULANA FAHRUDIN, S.S.T., MT ( 20219930501200 )  
SUGIARTO, S.KOM., M.KOM ( 19870214202 1211001 )

PROGRAM STUDI SAINS DATA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR  
2022

## **KATA PENGANTAR**

Segala puja dan puji syukur kehadiran Tuhan Yang Maha Esa., yang telah melimpahkan rahmat dan hidayah-Nya kepada kita semua, sehingga laporan akhir yang berjudul Laporan Akhir Project-Based Learning dapat diselesaikan pada waktunya. Walaupun demikian, penyusun berusaha dengan semaksimal mungkin demi kesempurnaan penyusunan laporan ini baik dari hasil kegiatan dalam perkuliahan. Saran dan kritik yang sifatnya membangun begitu diharapkan oleh penyusun demi kesempurnaan dalam penulisan laporan berikutnya.

Dalam kesempatan ini, penulis mengucapkan banyak terima kasih kepada semua pihak yang telah membantu dalam penyusunan Laporan Akhir Project-Based Learning ini, di antaranya:

1. Bapak Tresna Maulana Fahrudin, S.S.T., MT., selaku Dosen Pengampu Mata Kuliah Algoritma dan Pemrograman Lanjut
2. Rekan se-kelompok.

Akhir kata, penyusun berharap laporan ini dapat bermanfaat bagi pembaca. Saya ucapkan terima kasih banyak kepada semua pihak yang telah membantu, semoga Tuhan Yang Maha Esa membalas semua kebaikan kalian. Amin.

Surabaya, 09 Maret 2022

## DAFTAR ISI

<b>KATA PENGANTAR</b>	2
<b>DAFTAR ISI</b>	3
<b>BAB 1: PENDAHULUAN</b>	4
<b>1.1 Latar Belakang</b>	4
<b>1.2 Permasalahan</b>	5
<b>1.3 Tujuan</b>	6
<b>1.1 Manfaat</b>	6
<b>BAB II: TINJAUAN PUSTAKA</b>	7
<b>2.1 Teori Penunjang</b>	7
2.1.1. <i>Cyberbullying</i>	7
2.1.2. Analisis Sentimen	7
2.1.3. <i>Support vector machine</i>	7
2.1.4. <i>Text Preprocessing</i>	8
2.1.5. <i>Youtube</i>	9
<b>2.2 Penelitian Terkait</b>	9
<b>BAB III: METODOLOGI PENELITIAN</b>	12
<b>BAB IV: HASIL DAN PEMBAHASAN</b>	14
<b>4.1 Proses Analisis Data Komentar Pada Video Youtube</b>	14
4.1.1 Pengumpulan Data	14
4.1.2 Pre-processing data	15
4.1.3 Klasifikasi	20
4.1.4 Sentimen analisis	22
4.1.5 Mendesain UI sistem	24
4.1.6 Merubah file python menjadi <i>executable file</i>	28
4.1.7 Merubah folder ke file exe instalasi menggunakan aplikasi NSIS	29
4.1.8 Cara menjalankan cycek.exe	30
<b>4.2 Tingkat akurasi algoritma SVM</b>	32
<b>4.3 Visualisasi Word Cloud</b>	33
<b>BAB V: KESIMPULAN</b>	35
<b>DAFTAR PUSTAKA</b>	36
<b>LAMPIRAN</b>	38

# BAB 1: PENDAHULUAN

## 1.1 Latar Belakang

Media sosial adalah sebuah media online dengan para penggunanya bisa dengan mudah berpartisipasi, berbagi, dan menciptakan isi meliputi blog, jejaring sosial, wiki, forum dan dunia virtual. Media sosial juga bisa digunakan sebagai alat komunikasi jarak jauh, sehingga memudahkan pengguna ketika berinteraksi dengan orang lain tanpa harus bertemu secara langsung. Dengan hadirnya media sosial di Indonesia, tentu hal ini sangat digemari oleh masyarakat Indonesia. Aplikasi media sosial yang sering digunakan di Indonesia seperti instagram, twitter, tiktok, facebook dan lain-lain. Pengguna dari media sosial ini tak hanya dari kalangan orang dewasa saja, tetapi juga remaja dan anak-anak yang masih dalam bangku sekolah. Mereka dengan bebas bisa berekspresi di media sosial tanpa peduli dengan dampak yang didapat. Permasalahan utama yang kini terjadi di masyarakat kita melalui media sosial adalah tindakan bullying (*cyberbullying*).

Tindakan *cyberbullying* adalah salah satu dampak dari kebebasan berinteraksi di media sosial. Hal ini dilakukan dengan menggunakan media teks maupun gambar secara visual yang menjadi sarana komunikasi. Berdasarkan hasil riset microsoft, netizen Indonesia masuk ke dalam urutan ke-29 atau ketiga terendah sebagai netizen tidak sopan. Riset ini dilakukan di 32 negara dengan jumlah 16.000 responden dengan 503 netizen Indonesia. Faktor yang mempengaruhi penilaian ini adalah tindakan yang dilakukan ketika berselancar di dunia maya dan media sosial. Contoh tindakan *cyberbullying* yang paling sering dilakukan adalah penyebaran berita hoax dan penipuan sebanyak 47%, ujaran kebencian 27%, dan diskriminasi 13%. Berdasarkan kualifikasi usia, milenial dan generasi Z merupakan usia yang paling sering mengalami tindak *cyberbullying*. Usia kedua kelompok ini dimulai dari usia 41 tahun – 21 tahun, di mana golongan ini termasuk dalam kategori dewasa – anak muda. Yang dimana anak muda menjadi subjek utama dalam penyebaran kasus *cyberbullying*. Dilihat dari tingginya persentase jumlah anak muda yang mengalami kekerasan ini. Beberapa kasus yang terjadi juga menunjukkan bahwa korban bisa sekaligus menjadi pelaku. Oleh karena itu, Pemerintah dan anak muda adalah dua subjek yang paling berpengaruh untuk menurunkan tingkat kriminalitas dalam dunia digital.

Dengan analisis sentimen yang merupakan salah satu bidang dari Natural Language Processing (NLP), kita dapat meneliti permasalahan tersebut. Analisis

sentimen pada media sosial dilakukan dengan cara menyaring komentar-komentar dari aplikasi media sosial seperti instagram, tiktok dan twitter. Dari analisis tersebut dapat dilakukan tindakan preventif baik untuk korban maupun pelaku. Berbagai macam penelitian sosial mengenai *cyberbullying* telah dilakukan. Sebagai contohnya salah satu penelitian yang memberikan hasil bahwa tindakan bullying (perundungan) dapat dibagi menjadi dua yaitu, berupa mikro dan makro. Penelitian tersebut memberikan hasil bahwa beberapa diantaranya jejak dari pelaku bullying dapat diidentifikasi namun beberapa pelaku lainnya tidak dapat diidentifikasi karena menggunakan akun palsu (Nasrullah, 2015). Penelitian lain yang telah dilakukan ialah proses analisis sentimen yang dilakukan pada objek twitter dengan menerapkan metode Maximum Entropy dan Support Vector Machine (SVM). Berdasarkan penelitian tersebut hasil akurasi dengan mengimplementasikan metode SVM ialah 86,81% dengan waktu proses 1688 detik dan menggunakan 7 fold cross validation pada tipe kernel Sigmoid (Putranti & Winarko, 2014). Penelitian serupa membahas mengenai perbandingan hasil dengan mengimplementasikan metode Support Vector Machine-Particle Swarm Optimization dan metode yang hanya menggunakan SVM. Metode SVM dengan Particle Swarm Optimization (PSO) telah memberikan hasil akurasi sebesar 73,33%, sedangkan jika menerapkan algoritme SVM-PSO hasil akurasi mencapai 76% (Yunita, 2016). Penelitian mengenai metode SVM telah dilakukan pula dengan membandingkan hasil akurasi dengan metode Naïve Bayes dan Maximum Entropy. Berdasarkan penelitian tersebut didapatkan hasil akurasi terbaik ialah kurang lebih sebesar 98% pada metode SVM (K & Shetty, 2017).

Dari berbagai referensi penelitian yang telah dilakukan, metode Support Vector Machine menjadi salah satu pilihan metode yang akan digunakan dalam penelitian analisis sentimen. Hal ini menjadi topik pada penelitian ini untuk memberikan solusi terhadap *cyberbullying* yang terjadi di media sosial.

## **1.2 Permasalahan**

1. Bagaimana proses analisis sentimen *cyberbullying* pada komentar video youtube memanfaatkan algoritma *Support Vector Machine*?
2. Bagaimana tingkat akurasi yang didapatkan melalui algoritma *Support Vector Machine* terhadap analisis sentimen *cyberbullying* pada komentar video youtube?
3. Bagaimana hasil dari visualisasi word cloud yang telah dilakukan pada komentar video youtube?

### **1.3 Tujuan**

1. Mengetahui proses analisis sentimen *cyberbullying* pada komentar media sosial dengan memanfaatkan algoritma *Support Vector Machine*
2. Dapat mengetahui tingkat akurasi sentiment analysis melalui algoritma *Support Vector Machine*
3. Dapat menyajikan hasil dari visualisasi word cloud yang telah dilakukan pada komentar video youtube

### **1.1 Manfaat**

Adapun manfaat dari penelitian ini bagi penulis yaitu untuk melatih pemahaman tentang analisis sentimen. Selain itu penulis juga dapat mempelajari hal baru yang di dapat dari adanya projek ini. Sedangkan manfaat bagi masyarakat yaitu mereka menjadi lebih tau manakah konten yang menimbulkan banyak *cyberbullying*, dan komentar seperti apakah yang masuk dalam perilaku *cyberbullying*. Selain itu mereka juga dapat mendapat ilmu baru tentang analisis sentimen.

## BAB II: TINJAUAN PUSTAKA

### 2.1 Teori Penunjang

#### 2.1.1. *Cyberbullying*

*Cyberbullying* adalah tindakan negatif yang dilakukan oleh seseorang atau kelompok tertentu dengan cara mengirimkan pesan teks, foto, gambar meme, dan video ke akun media sosial seseorang dengan tujuan untuk menyindir, menghina, melecehkan, mendiskriminasi bahkan mempersekusi individu [11]. Motif seseorang melakukan tindakan *cyberbullying* adalah ketidaksukaan terhadap seseorang atau pribadi seseorang, bermaksud menyindir dengan kalimat-kalimat negatif yang kurang etis dan kasar, bertujuan untuk menghibur agar para user atau pengguna internet dapat tertawa. Namun, hal tersebut bisa saja dapat memberikan dampak negatif bagi psikologis seseorang.

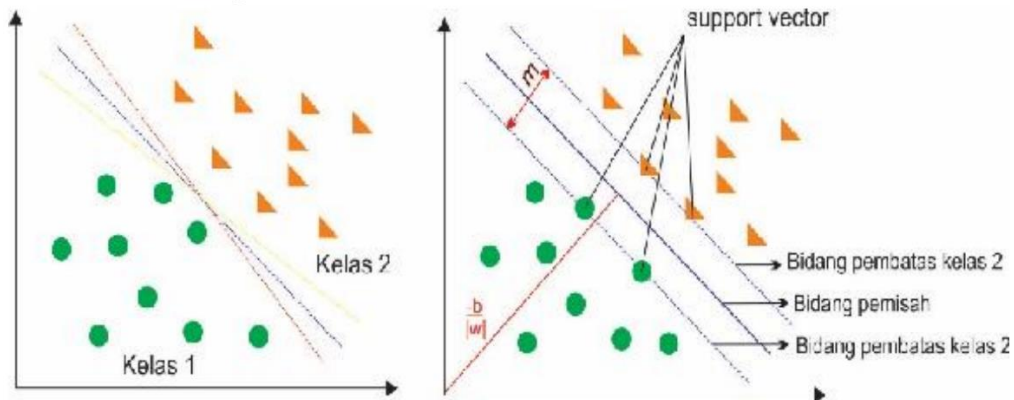
#### 2.1.2. Analisis Sentimen

Analisis sentimen adalah proses mengekstraksi, memahami, dan mengolah data berupa text yang tidak terstruktur secara otomatis guna mendapatkan informasi sentiment yang terdapat pada sebuah kalimat pendapat atau opini [2]. Analisis sentimen sering kali disebut sebagai *opinion mining*, karena nantinya kita akan menggali emosi apa saja yang ada dibalik sebuah text. Analisis sentimen memiliki empat tipe, yaitu *fine-grained sentiment analysis*, *emotion detection*, *aspect-based sentiment analysis*, dan *multilingual sentiment analysis*. Analisis sentimen dapat digunakan untuk memonitoring media sosial, memonitoring *brand*, *customer feedback*, *customer service*, ataupun untuk melihat riset pasar.

#### 2.1.3. *Support vector machine*

*Support Vector Machine* (SVM) adalah salah satu algoritma machine learning dengan pendekatan supervised learning yang bekerja dengan mencari hyperplane atau fungsi pemisah terbaik untuk memisahkan kelas [6]. *Support Vector Machine* merupakan salah satu metode yang biasa digunakan untuk klasifikasi dan regresi pada data mining. SVM dapat mengatasi masalah klasifikasi dan regresi baik dengan linear maupun non linear. SVM digunakan untuk mencari hyperplane terbaik dengan memaksimalkan jarak antar kelas.

Adapun hyperplane adalah sebuah fungsi yang digunakan sebagai pemisah antar kelas. Support Vector Machine dapat memisahkan data secara linier dengan 2 variabel  $x$  sebagai dataset dan  $y$  sebagai label untuk dataset. Berikut merupakan penggambaran Support Vector Machine dengan kasus linier :



Gambar 1. Penggambaran Support Vector Machine

Sumber : Kurniawan, Taufik, 2017

Data kelas atau label akan terpisah oleh garis lurus sejajar, oleh karena itu ini disebut Support Vector Machine Linier Seperable Data. Sedangkan data diantara bidang pembatas disebut dengan Support Vector. Hyperplane berguna dalam memisahkan 2 kelompok class +1 dan class -1 dimana setiap class memiliki pola masing-masing (Luqyana, Cholissodin and Perdana, 2018). Bidang pertama membatasi class pertama sedangkan bidang kedua membatasi class kedua.

#### 2.1.4. Text Preprocessing

*Text preprocessing* merupakan proses pembersihan data untuk mendapatkan data yang jauh lebih bagus. Adapun tahapan dalam *text preprocessing* meliputi *case folding*, *tokenizing*, *filtering*, dan *stemming* [7]. Pada tahapan *case folding*, dilakukan penyamarataan huruf. Untuk karakter lain yang bukan termasuk huruf dan angka dianggap sebagai delimiter dan akan dihapus. Pemecahan kalimat menjadi kata-kata masuk dalam tahap *tokenizing*. Lanjutan dari tahapan *tokenizing* adalah tahapan *filtering* yang digunakan untuk mengambil kata-kata yang penting dari hasil token tadi. Kata umum yang biasanya muncul dan tidak memiliki makna disebut dengan *stopword*. Tahap *stemming* yang merupakan tahap terakhir yaitu tahapan yang juga diperlukan



untuk memperkecil jumlah indeks yang berbeda dari satu data sehingga sebuah kata yang memiliki suffix maupun prefix akan kembali ke bentuk dasarnya.

#### 2.1.5. Youtube

*Youtube* merupakan salah satu jenis situs jejaring sosial yang memberikan fasilitas visual dan suara kepada pengguna. *Youtube* saat ini banyak sekali digemari oleh anak muda. Hal ini dikarenakan dapat melihat secara langsung visualisasi bergerak. *Youtube* merupakan database video yang paling populer di dunia internet, dan merupakan situs video yang menyediakan berbagai informasi berupa gambar bergerak dan bisa diandalkan [13]. Situs ini memang disediakan bagi mereka yang ingin melakukan pencarian informasi video dan menontonnya secara langsung.

## 2.2 Penelitian Terkait

2.2.1 Penelitian yang dilakukan oleh Primandani Arsi dan Retno Waluyo, mahasiswa informatika dan sistem informasi Universitas Amikom Purwokerto yang dilakukan pada tahun 2020 dengan judul “ANALISIS SENTIMEN WACANA PEMINDAHAN IBU KOTA INDONESIA MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM)”. Dalam penelitian ini diusulkan metode Support Vector Machine(SVM) untuk diterapkan pada *tweets* topik pemindahan ibukota Indonesia untuk tujuan klasifikasi kelas sentimen pada media sosial twitter. Teknis klasifikasi dilakukan dengan cara mengklasifikasikan menjadi 2 kelas yakni positif dan negatif. Berdasarkan hasil pengujian yang dilakukan terhadap *tweets* sentimen pemindahan ibukota dari media sosial twitter sebanyak 1.236 *tweets*. Adapun akurasinya sebagai berikut :

☒ Table View 
 ☐ Plot View

accuracy: 96.68% +/- 0.60% (micro average: 96.68%)

	true Negative	true Positive	class precision
pred. Negative	815	24	97.14%
pred. Positive	17	380	95.72%
class recall	97.96%	94.06%	

Gambar 2. Evaluasi Penelitian Terkait I

Hasil evaluasi yang didapatkan melalui model SVM menghasilkan Confusion Matrix dengan akurasi = 96,68%, precision = 95.82%, recall = 94.04% dan AUC = 0,979.

2.2.2 Penelitian yang dilakukan oleh Wanda Athira Luqyana, mahasiswa Program Studi Teknik Informatika Universitas Brawijaya yang dilakukan pada tahun

2018 dengan judul “ANALISIS SENTIMEN *CYBERBULLYING* PADA KOMENTAR INSTAGRAM DENGAN METODE KLASIFIKASI SUPPORT VECTOR MACHINE”. Penelitian ini mengambil dokumen dengan rincian 400 data diambil secara luring dengan total fitur 1799. Dokumen komentar dibagi menjadi 70% data latih dan 30% data uji. Berdasarkan pengujian yang dilakukan didapatkan parameter terbaik pada metode SVM yaitu dengan nilai degree kernel polynomial sebesar 2, nilai learning rate sebesar 0,0001, dan jumlah iterasi maksimum yang digunakan adalah 200 kali. Dari pengujian tersebut didapatkan hasil akurasi tertinggi sebesar 90% pada komposisi data latih 50% dan komposisi data uji 50%.

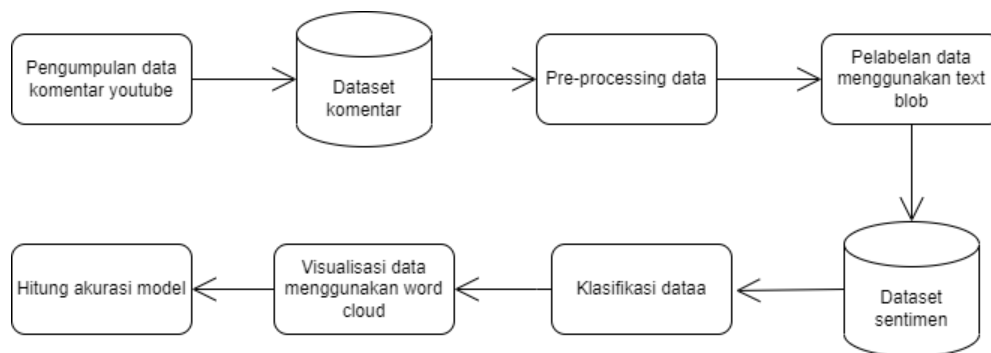
- 2.2.3 Penelitian yang dilakukan oleh Fransiska Vina Sari dan Arief Wibowo, mahasiswa Magister Ilmu Komputer Fakultas Teknologi Informasi Universitas Budi Luhur, Jakarta dilakukan pada tahun 2019 dengan judul “ANALISIS SENTIMEN PELANGGAN TOKO ONLINE JD.ID MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER BERBASIS KONVERSI IKON EMOSI”. Penelitian ini menggunakan metode Naïve Bayes Classifier (NBC) dengan pembobotan tf-idf disertai penambahan fitur konversi ikon emoji (emoticon) untuk mengetahui kelas sentimen yang ada dari tweet tentang toko JD.id. Hasil penelitian menunjukkan bahwa metode Naïve Bayes tanpa penambahan fitur mampu mengklasifikasi sentimen dengan nilai akurasi sebesar 96,44%, sementara jika ditambahkan fitur pembobotan tf-idf disertai konversi ikon emosi mampu meningkatkan nilai akurasi menjadi 98%.
- 2.2.4 Penelitian yang dilakukan oleh Mediana Saraswati dan Desti Rimirasih mahasiswa Program Studi Teknik Informatika Universitas Gunadarma pada tahun 2020 dengan judul “ANALISIS SENTIMEN TERHADAP PELAYANAN KRL COMMUTERLINE BERDASARKAN DATA TWITTER MENGGUNAKAN ALGORITMA BERNOULLI NAIVE BAYES”. Penelitian ini bertujuan untuk mengklasifikasikan tweets mengenai pelayanan KRL Commuterline ke dalam sentimen positif dan negatif dengan pendekatan machine learning menggunakan algoritma Bernoulli Naive Bayes. Data latih yang digunakan sebanyak 2.690 data dan data validasi sebanyak 1.626 data tweets. Pada proses pelatihan model diperoleh tingkat akurasi pelatihan sebesar 86,36% dan tingkat akurasi validasi sebesar 85,73%.

Pengujian model dilakukan menggunakan 20 tweets baru dan diperoleh tingkat akurasi sebesar 85%.

- 2.2.5 Penelitian yang dilakukan oleh Ulfa Khaira, Ragil Johanda, Pradita Eko Prasetyo Utomo, dan Tri Suratno dosen Program Studi Sistem Informasi Universitas Jambi pada tahun 2020 dengan judul “SENTIMENT ANALYSIS OF CYBERBULLYING ON TWITTER USING SENTISTRENGTH”. Penelitian ini menggunakan SentiStrength, sebuah algoritma yang menggunakan pendekatan berbasis leksikon. Leksikon SentiStrength ini mengandung bobot sentimennya kekuatan. Hasil penilaian dari 454 data tweet diperoleh 161 tweet non-intimidasi (35,4%), 87 tweet netral (19,1%), dan 206 tweet intimidasi (45,4%). Penelitian ini menghasilkan nilai akurasi 60,5%.

### BAB III: METODOLOGI PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah metode eksperimen. Penelitian ini memanfaatkan *support vector machine* untuk melakukan analisis terhadap komentar pada video youtube dengan melakukan pelabelan sebagai komentar positif, negatif, atau natural. Dalam hal ini, penelitian akan melalui enam tahapan yaitu pengumpulan data komentar YouTube, preprocessing data, labelling menggunakan text blob, klasifikasi komentar YouTube, visualisasi word cloud dan menghitung akurasi model seperti yang tampak pada diagram sistem berikut:



Gambar 3. Desain sistem yang diusulkan dalam project

Peneliti menggunakan metode pemilihan sampel berupa metode Purposivesampling atau sebuah teknik untuk menentukan sampel melalui pertimbangan tertentu yang dianggap sesuai dengan tujuan penelitian secara sengaja. Sampel yang digunakan dalam penelitian ini adalah data komentar video yang didapatkan dalam *Youtube* melalui bantuan spreadsheet dengan hasil berupa file berekstensi csv.

Untuk dapat melakukan pengembangan informasi perlu dilakukan beberapa tahapan yang harus dilakukan dengan tujuan mengolah sumber data menjadi data yang baik dan terstruktur, tahapan yang telah dimaksud ialah proses pre-processing data yang terdiri dari :

#### 1. Case folding

*Case folding* merupakan proses dalam text preprocessing yang dilakukan untuk menyeragamkan karakter pada data. Proses *case folding* adalah proses mengubah seluruh huruf menjadi huruf kecil seperti ‘Apakah makanan yang dirasakan enak?’ menjadi ‘apakah makanan yang dirasakan enak?’.

#### 2. Cleansing

Proses yang kedua adalah *cleansing* berupa pembersihan data dari simbol, angka, username, hashtag, tanda baca, url dan lainnya yang dirasa dapat mengganggu proses ekstraksi fitur dan klasifikasi komentar.

### 3. *Stopword*

Data juga akan mengalami proses stopword, dalam proses ini kata sambung atau yang tidak relevan akan dihapus dan dihilangkan. Misalnya seperti kata tetapi, untuk, dengan, yang, pada, dan kata sambung lainnya.

### 4. *Tokenize*

Selanjutnya masuk dalam proses *tokenize*, merupakan penguraian kalimat menjadi token-token atau kata tunggal. Contohnya seperti polisi telah menerapkan ganjil genap menjelang libur lebaran, dan setelah ditokenizing maka menjadi ‘polisi’, ‘telah’, ‘menerapkan’, ‘ganjil’, ‘genap’, ‘menjelang’, ‘libur’, ‘lebaran’.

### 5. Normalisasi kata

Setelah melalui proses *tokenize*, kata-kata yang diperoleh tidak selalu dalam bahasa yang baik dan akan terdapat kata-kata yang ambigu. Oleh karena itu, data tersebut akan melalui proses normalisasi untuk mengubah tiap-tiap kata menjadi kata baku yang mudah dipahami.

### 6. *Stemming*

Proses terakhir dari pre-processing data adalah proses mengubah kata yang berimbuhan menjadi kata dasar dengan menghilangkan imbuhan awalan atau akhiran seperti me, meny, meng, di, per, ber, an, kan, i, nya, dan sebagainya.

Sebelum masuk ke tahap pelabelan data, data yang telah dihasilkan dari pre-processing akan diterjemahkan ke dalam bahasa inggris. Data tersebut kemudian diubah menjadi blob atau kode biner. Kemudian data yang dihasilkan akan dihitung polaritasnya. Dari hasil polaritas, data tersebut akan diberi nilai dan terdapat interval tertentu yang digunakan sebagai acuan bahwa komentar termasuk negatif, positif, atau netral.

Dalam penelitian ini juga perlu mengetahui kata apa yang dominan dalam komentar video youtube menggunakan metode visualisasikan word cloud. Pada proses terakhir, peneliti membutuhkan langkah berupa mengukur akurasi model dari metode *support vector machine* yang telah dilakukan dengan akurasi dan presisi linear serta akurasi dan presisi RBF.

## BAB IV: HASIL DAN PEMBAHASAN

### 4.1 Proses Analisis Data Komentar Pada Video Youtube

#### 4.1.1 Pengumpulan Data

Pada penelitian ini, kami mengumpulkan data komentar YouTube untuk dilakukan analisis. Adapun caranya adalah sebagai berikut:

- A. Membuat blank document di spreadsheet
- B. Ambil id video YouTube yang akan di download komentarnya
- C. Copy id video di kotak pertama spreadsheet
- D. Tekan tools dan klik script editor
- E. Copy script koding dan jalankan

```
function scrapeCommentsWithoutReplies(){
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var
  result=[['Name','Comment','Time','Likes','Reply
Count']];
  var vid =
ss.getSheets()[0].getRange(1,1).getValue();
  var nextPageToken=undefined;

  while(1){
    var data =
YouTube.CommentThreads.list('snippet', {videoId:
vid, maxResults: 100, pageToken: nextPageToken})
    nextPageToken=data.nextPageToken
    //console.log(nextPageToken);
    for (var row=0; row<data.items.length; row++) {

result.push([data.items[row].snippet.topLevelComment
.snippet.authorDisplayName,

data.items[row].snippet.topLevelComment.snippet.text
Display,

data.items[row].snippet.topLevelComment.snippet.publ
ishedAt,

data.items[row].snippet.topLevelComment.snippet.like
Count,

data.items[row].snippet.totalReplyCount]);
    }
    if(nextPageToken =="" || typeof nextPageToken
=== "undefined"){
```

```

        break;
    }
}
var newSheet=ss.insertSheet(ss.getNumSheets())
newSheet.getRange(1,
1,result.length,5).setValues(result)

}

```

*Gambar 4. Script untuk scrapping komentar YouTube*

#### F. Cek di spreadsheet dan komentar sudah tersedia

Setelah melalui seluruh proses tersebut, data yang dikumpulkan berupa data komentar video youtube yang didapatkan dari channel KOMPASTV dengan judul “Video Amatir Detik-Detik Seorang Pelawak Ditangkap Polisi Karena Narkoba” dalam link berikut: <https://www.youtube.com/watch?v=bDdhqh4QzM>. Keseluruhan total komentar ialah 4.600 dan didapatkan data sejumlah 3.777 komentar berupa data tabel dalam file berekstensi csv.

#### 4.1.2 Pre-processing data

```

#import library
import pandas as pd
import numpy as np
import nltk
import string
import csv
import re
from Sastrawi.Stemmer.StemmerFactory import
StemmerFactory
import warnings
warnings.filterwarnings("ignore")

class process:
    def __init__(self, file):
        self.file = pd.read_csv(file)

    def baca(self):
        komena = self.file
        return komena

    def simpan(self):
        self.file.to_excel('data_2_inggris.xlsx')

    def case_folding(self):
        self.file['Comment']=
self.file['Comment'].str.lower()

```

```

def remove_number(self, text):
    def remove_number(text):
        return re.sub(r"\d+", "", text)
    self.file['Comment'] =
self.file['Comment'].apply(remove_number)

def remove_punemot(self, text): #punctuation
& emoticon
    self.file['Comment']=
self.file['Comment'].str.replace('[^\w\s]','')

def remove_url(self, text):
    def remove_URL(sample):
        return re.sub(r"http\S+", "", sample)
    self.file['Comment'] =
self.file['Comment'].apply(remove_URL)

def stopword(self, text):
    stopword =
nltk.corpus.stopwords.words('indonesian')

    stopword.extend(['dehhh', 'wkwk',
'busukin', 'tersesat', 'mampus', 'aura',
'kesel', 'ampun', 'lu', 'loe', 'dehhh', 'la',
'eh', 'awokawoak', 'cu',
'hehehehe', 'lah', 'yap', 'paan', 'cui',
'cak', 'yuuuk', 'coki'])

    txt_stopword = pd.read_csv('stopwords-
id.txt', names= ['stopwords'], header= None)

stopword.extend(txt_stopword['stopwords'][0].spli
t(' '))

    stopword = set(stopword)

    def stopwords(text):
        text= [word for word in text if word
not in stopword]
        return text

    self.file['Comment']=
self.file['Comment'].apply(stopwords)

def token(self, text):
    def token(text):
        text=re.split('\W+',text)
        return text
    self.file['Comment'] =
self.file['Comment'].apply(token)

```



```

        def normalisasi(self, text):
            normalized_word =
pd.read_csv('normalisasi.csv')

            normalized_word_dict={}

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

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

            self.file['Comment'] =
self.file['Comment'].apply(normalized_term)

            def stemd(self, text):
                factory = StemmerFactory()
                stemmer = factory.create_stemmer()

            def stemmed_wrapper(term):
                return stemmer.stem(term)
            term_dict= {}

            for document in self.file['Comment']:
                for term in document:
                    if term not in term_dict:
                        term_dict[term] = ' '

            for term in term_dict:
                term_dict[term] =
stemmed_wrapper(term)

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

            self.file['Comment'] =
self.file['Comment'].apply(get_stemmed_term)

            def fitt_stopword(self, text):
                def fit_stopwords(text):
                    text= np.array(text)
                    text= ' '.join(text)
                    return(text)

```

```

        self.file['Comment']=
self.file['Comment'].apply(lambda x:
fit_stopwords(x))

```

Gambar 5. Potongan Script pada Proses Preprocessing Data

### 1. Case folding

Tahap *Case folding* menghasilkan data komentar yang menggunakan huruf kecil pada kesemuanya.

No	Sebelum Case folding	Setelah Case folding
1	Semua org yg klopun gak salah sekalipun klo tiba2 digrebek juga pasti panik lah, ada2 aja yg bikin caption	semua org yg klopun gak salah sekalipun klo tiba2 digrebek juga pasti panik lah, ada2 aja yg bikin caption
2	Gila tapi dia sendiri kan yg serahin bb nya	gila tapi dia sendiri kan yg serahin bb nya
3	Tersangka kalau dah korpuratif putagas gk perlu bentak2..itulah penyeba rakyat banyak yg gk suka sama polisi..terlalu arogan..	tersangka kalau dah korpuratif putagas gk perlu bentak2..itulah penyeba rakyat banyak yg gk suka sama polisi..terlalu arogan..
4	C*li Pardede di Penjara Ga C*li Lagi Dong Hahhhaaaaa...	c*li pardede di penjara ga c*li lagi dong hahhhaaaaa...

Tabel 1. hasil case folding

### 2. Cleansing

Tampilan data setelah mengalami proses *case folding* dan juga *cleansing* dengan menghilangkan komponen tertentu seperti simbol, angka, username, hashtag, tanda baca, url dan lainnya ditunjukkan pada tabel di bawah ini.

No	Sebelum Cleansing	Setelah Cleansing
1	semua org yg klopun gak salah sekalipun klo tiba2 digrebek juga pasti panik lah, ada2 aja yg bikin caption	semua org yg klopun gak salah sekalipun klo tiba digrebek juga pasti panik lah ada aja yg bikin caption
2	gila tapi dia sendiri kan yg serahin bb nya	gila tapi dia sendiri kan yg serahin bb nya
3	tersangka kalau dah korpuratif putagas gk perlu bentak2..itulah penyeba rakyat banyak yg gk suka sama polisi..terlalu arogan..	tersangka kalau dah korpuratif putagas gk perlu bentakitulah penyeba rakyat banyak yg gk suka sama polisiterlalu arogan
4	c*li pardede di penjara ga c*li lagi dong hahhhaaaaa...	cli pardede di penjara ga cli lagi dong hahhhaaaaa

Tabel 2. hasil cleansing

### 3. Tokenize

Setelah melalui tahap *tokenizing*, data telah berubah menjadi kata dengan batasan tanda baca dan spasi. simulasi hasil tokenizing dapat dilihat pada tabel dibawah ini.

No	Sebelum Tokenizing	Setelah Tokenizing
1	semua org yg klopun gak salah sekalipun klo tiba digrebek juga pasti panik lah ada aja yg bikin caption	['semua', 'org', 'yg', 'klopun', 'gak', 'salah', 'sekalipun', 'klo', 'tiba', 'digrebek', 'juga', 'pasti', 'panik', 'lah', 'ada', 'aja', 'yg', 'bikin', 'caption']
2	gila tapi dia sendiri kan yg serahin bb nya	['gila', 'tapi', 'dia', 'sendiri', 'kan', 'yg', 'serahin', 'bb', 'nya']
3	tersangka kalau dah korpuratif putagas gk perlu bentakitulah penyeba rakyat banyak yg gk suka sama polisiterlalu arogan	['tersangka', 'kalau', 'dah', 'korporatif', 'putagas', 'gk', 'perlu', 'bentakitulah', 'penyeba', 'rakyat', 'banyak', 'yg', 'gk', 'suka', 'sama', 'polisiterlalu', 'arogan']
4	cli pardede di penjara ga cli lagi dong hahhhaaaaa	['cli', 'pardede', 'di', 'penjara', 'ga', 'cli', 'lagi', 'dong', 'hahhhaaaaa']

Tabel 3. hasil tokenizing

### 4. Normalisasi kata

Berikut tampilan tiap-tiap kata setelah mengalami normalisasi kata dan berubah menjadi kata baku.

No	Sebelum Normalisasi Kata	Setelah Normalisasi Kata
1	['semua', 'org', 'yg', 'klopun', 'gak', 'salah', 'sekalipun', 'klo', 'tiba', 'digrebek', 'juga', 'pasti', 'panik', 'lah', 'ada', 'aja', 'yg', 'bikin', 'caption']	['semua', 'orang', 'yang', 'klopun', 'tidak', 'salah', 'sekalipun', 'kalau', 'tiba', 'digrebek', 'juga', 'pasti', 'panik', 'lah', 'ada', 'saja', 'yang', 'bikin', 'caption']
2	['gila', 'tapi', 'dia', 'sendiri', 'kan', 'yg', 'serahin', 'bb', 'nya']	['gila', 'tetapi', 'dia', 'sendiri', 'kan', 'yang', 'serahin', 'bb', 'nya']
3	['tersangka', 'kalau', 'dah', 'korporatif', 'putagas', 'gk', 'perlu', 'bentakitulah', 'penyeba', 'rakyat', 'banyak', 'yg', 'gk', 'suka', 'sama', 'polisiterlalu', 'arogan']	['tersangka', 'kalau', 'dah', 'korporatif', 'putagas', 'tidak', 'perlu', 'bentakitulah', 'penyeba', 'rakyat', 'banyak', 'yang', 'tidak', 'suka', 'sama', 'polisiterlalu', 'arogan']
4	['cli', 'pardede', 'di', 'penjara', 'ga', 'cli', 'lagi', 'dong', 'hahhhaaaaa']	['cli', 'pardede', 'di', 'penjara', 'tidak', 'cli', 'lagi', 'dong', 'hahhhaaaaa']

Tabel 4.hasil normalisasi

### 5. Stopword

Berikut ialah output setelah data melewati proses *stopword* dan telah hilang kata-kata yang tidak penting atau tidak dibutuhkan seperti kata sambung.

No	Sebelum Stopword	Setelah Stopword
1	['semua', 'orang', 'yang', 'klopun', 'tidak', 'salah', 'sekalipun', 'kalau', 'tiba', 'digrebek', 'juga', 'pasti', 'panik', 'lah', 'ada', 'saja', 'yang', 'bikin', 'caption']	['orang', 'klopun', 'salah', 'digrebek', 'panik', 'bikin', 'caption']
2	['gila', 'tetapi', 'dia', 'sendiri', 'kan', 'yang', 'serahin', 'bb', 'nya']	['gila', 'serahin', 'bb', 'nya']
3	['tersangka', 'kalau', 'dah', 'korporatif', 'putagas', 'tidak', 'perlu', 'bentakitulah', 'penyeba', 'rakyat', 'banyak', 'yang', 'tidak', 'suka', 'sama', 'polisiterlalu', 'arogan']	['tersangka', 'dah', 'korporatif', 'putagas', 'bentakitulah', 'penyeba', 'rakyat', 'suka', 'polisiterlalu', 'arogan']
4	['cli', 'pardede', 'di', 'penjara', 'tidak', 'cli', 'lagi', 'dong', 'hahhhaaaaa']	['cli', 'pardede', 'penjara', 'cli', 'hahhhaaaaa']

Tabel 5. hasil stopwords

## 6. Stemming

Berikut ialah hasil akhir dari data setelah melalui proses terakhir pre-processing data yang menghasilkan kata dasar pada keseluruhannya.

No	Sebelum Stemming	Setelah Stemming
1	['orang', 'klopun', 'salah', 'digrebek', 'panik', 'bikin', 'caption']	['orang', 'klopun', 'salah', 'digrebek', 'panik', 'bikin', 'caption']
2	['gila', 'serahin', 'bb', 'nya']	['gila', 'serahin', 'bb', 'nya']
3	['tersangka', 'dah', 'korporatif', 'putagas', 'bentakitulah', 'penyeba', 'rakyat', 'suka', 'polisiterlalu', 'arogan']	['sangka', 'dah', 'korporatif', 'putagas', 'bentakitulah', 'seba', 'rakyat', 'suka', 'polisiterlalu', 'arogan']
4	['cli', 'pardede', 'penjara', 'cli', 'hahhhaaaaa']	['cli', 'pardede', 'penjara', 'cli', 'hahhhaaaaa']

Tabel 6. hasil stemming

### 4.1.3 Klasifikasi

```

from textblob import TextBlob
import pandas as pd

class klasifikasi():

    def __init__(self, text):
        self.file = pd.read_csv(text)

    def baca(self):
        komena = self.file
        return komena

    def simpan(self):

```

```

        self.file.to_csv('data_2_inggris_class.csv')

    def textblob(self):
        for yt in self.file.Text_English:
            clean = yt

            blob_object = TextBlob (clean)
            hasil = blob_object.tags
            print(hasil)

    def polarity(self, text):
        polarityy = lambda x:
TextBlob(x).sentiment.polarity

        self.file['Polarity'] =
self.file['Text_English'].apply(polarityy)

    def analysis(self, text):
        def analysis(score):
            if score > 0 :
                return 'positive'
            elif score == 0:
                return 'neutral'
            else:
                return 'negative'
        self.file['Score']=
self.file['Polarity'].apply(analysis)

    def hitung(self, text):
        print('netral :',
(sum(self.file['Score']=='neutral')))
        print('positif :',
(sum(self.file['Score']=='positive')))
        print('negatif :',
(sum(self.file['Score']=='negative')))

polar = dataset2.polarity(blob)
analysis = dataset2.analysis(polar)
htng = dataset2.hitung(analysis)
baca= dataset2.baca()
print(baca)

```

*Gambar 6. Potongan script pada tahap klasifikasi*

Setelah dilakukan tahap klasifikasi, tiap data telah diubah menjadi kode biner dan telah dihitung polaritasnya. Melalui perhitungan polaritas telah diketahui data komentar masih dalam kategori negatif, positif, atau netral. Pada data komentar YouTube yang berjudul “Video Amatir Detik-Detik Seorang Pelawak Ditangkap Polisi Karena Narkoba” menunjukkan bahwa komentar positif sebanyak 737, netral

1.998, dan negatif 341. Dengan jumlah komentar netral tertinggi, maka dapat disimpulkan bahwa tingkat cyberbullying pada komentar video tersebut rendah.

#### 4.1.4 Sentimen analisis

```
import pandas as pd
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import
CountVectorizer
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import cross_val_score

class analisis:
    def __init__(self, file):
        self.file = pd.read_csv(file)

    def baca(self):
        komena = self.file
        return komena

    def wc(self):
        text = ' '.join(yt for yt in
self.file['Comment'])

        WordCloud = WordCloud(width = 600, height =
300, random_state = 21, max_font_size = 200,
collocations = False).generate(text)
        plt.figure(figsize = (20,10))
        plt.imshow(WordCloud, Interpolation =
'bilinear')
        plt.axis('off')

    def createdatatrain(self):
        self.file['Text_English']=
self.file['Text_English'].astype(str)
        y= self.file['Score'].values
        x= self.file['Text_English'].values
        self.file.x_train, self.file.x_test,
self.file.y_train, self.file.y_test =
train_test_split(x, y, stratify=y, random_state=1,
test_size=0.2, shuffle= True)
        print(self.file.x_train.shape)
        print(self.file.x_test.shape)
```

```

        def hitung(self, text):
            vectorizer = CountVectorizer(binary= True,
stop_words = 'english')
            vectorizer.fit (list(self.file.x_train)+
list(self.file.x_test))
            self.file.x_train_vec =
vectorizer.transform(self.file.x_train)
            self.file.x_test_vec =
vectorizer.transform(self.file.x_test)
            print(self.file.x_train_vec.shape)
            print(self.file.x_test_vec.shape)

        def linear(self, text):
            svm_linear= svm.SVC(kernel='linear',
probability= True, C=100)
            prob= svm_linear.fit(self.file.x_train_vec,
self.file.y_train).predict_proba(self.file.x_test_vec)

            y_pred_svm_linear=
svm_linear.predict(self.file.x_test_vec)

            print("Accuracy Linear :",
accuracy_score(self.file.y_test,
y_pred_svm_linear)*100, '%')
            scores = cross_val_score(svm_linear,
self.file.x_train_vec, self.file.y_train, cv=10)

            print('\n')
            print('confusion matrix linear :')
            print(confusion_matrix(self.file.y_test,
y_pred_svm_linear))

            print('\n')
            print('precision linear :',
precision_score(self.file.y_test, y_pred_svm_linear,
average='macro'))

            print('\n')
            print('recall linear :',
recall_score(self.file.y_test, y_pred_svm_linear,
average='macro'))

            print('\n')
            print('cross validation linear :', scores)

        def rbf(self, text):
            rbf= svm.SVC(kernel='rbf', probability= True,
C=100, gamma= 0.01 )
            prob= rbf.fit(self.file.x_train_vec,
self.file.y_train).predict_proba(self.file.x_test_vec)

```

```

        y_pred_svm_rbf=
rbf.predict(self.file.x_test_vec)

        print("Accuracy RBF :",
accuracy_score(self.file.y_test,
y_pred_svm_rbf)*100,'%')
        scores = cross_val_score(rbf,
self.file.x_train_vec, self.file.y_train, cv=10)

        print('\n')
        print('confusion matrix rbf :')
        print(confusion_matrix(self.file.y_test,
y_pred_svm_rbf))

        print('\n')
        print('precision rbf :',
precision_score(self.file.y_test, y_pred_svm_rbf,
average='macro'))

        print('\n')
        print('recall rbf :',
recall_score(self.file.y_test, y_pred_svm_rbf,
average='macro'))

        print('\n')
        print('cross validation rbf :', scores)

data = analisis('data_2_inggris_class.csv')

train = data.createdatatrain()
htg = data.hitung(train)
linear = data.linear(htg)
rbf = data.rbf(htg)

```

*Gambar 7. Potongan script pada sentimen analisis*

Setelah melakukan proses ini dapat diketahui tingkat akurasi metode SVM dan juga hasil visualisasi Word Cloud dari data komentar video youtube berjudul “Video Amatir Detik-Detik Seorang Pelawak Ditangkap Polisi Karena Narkoba”.

#### **4.1.5 Mendesain UI sistem**

```

pip install google-trans-new

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import svm

```



```

from sklearn.metrics import classification_report
from google_trans_new import google_translator
from tkinter import *
from tkinter import ttk
from tkinter import messagebox

data = pd.read_csv('data_2_inggris_class.csv')
df = data
df.head()
data = df.fillna(' ')
y= data.Score.values
x= data.Text_English.values

x_train, x_test, y_train, y_test = train_test_split(x, y,
stratify=y, random_state=1, test_size=0.2, shuffle= True)
print(x_train.shape)
print(x_test.shape)

vectorizer = CountVectorizer(binary= True, stop_words
='english')
vectorizer.fit(list(x_train)+list(x_test))

x_train_vec = vectorizer.transform(x_train)
x_test_vec = vectorizer.transform(x_test)
print(x_train_vec.shape)
print(x_test_vec.shape)

rbf= svm.SVC(kernel='rbf', probability= True, C=100,
gamma= 0.01 )
prob= rbf.fit(x_train_vec,
y_train).predict_proba(x_test_vec)

y_pred_svm_rbf= rbf.predict(x_test_vec)

translator = google_translator()
root = Tk()
root.title("Analisis Sentimen Cyberbullying")
frame_header = ttk.Frame(root)
frame_header.pack()
logo = PhotoImage(file='logo.png',
master=frame_header).subsample(2, 2)
logolabel = ttk.Label(frame_header, text='logo',
image=logo)
logolabel.grid(row=0, column=0, rowspan=2)

headerlabel = ttk.Label(frame_header, text='CYCEK
(Cyberbullying Check)', font=('Arial', 18, 'bold'))
headerlabel.grid(row=0, column=1)

```

```

messagelabel = ttk.Label(frame_header, wraplength=300,
text='Sistem cyberbullying check ini berfungsi sebagai
sarana pengecekan apakah kalimat yang kalian tuliskan
termasuk dalam kalimat bullying yang negatif, positif,
atau netral.')
messagelabel.grid(row=1, column=1)
frame_content = ttk.Frame(root)
frame_content.pack()
# def submit():
# username = entry_name.get()
# print(username)
myvar = StringVar()
var = StringVar()
# cmnt= StringVar()
commentlabel = ttk.Label(frame_content, text='Input Your
Comment: ', font=('Arial', 10))
commentlabel.grid(row=0, column=0, sticky='sw')
textcomment = Text(frame_content, width=55, height=10)
textcomment.grid(row=1, column=0, padx=5, columnspan=2)

textcomment.config(wrap = 'word')
# def clear():
# textcomment.delete(1.0, 'end')
def clear():
    global textcomment
    messagebox.showinfo(title='Clear', message='Do you
want to clear?')
    textcomment.delete(1.0, END)

def submit():
    global textcomment
    text_masuk = str(textcomment.get(1.0, END)) #untuk
menyimpan text yang ada di entry box
    translate_text = translator.translate(text_masuk,
lang_tgt='en')
    y_pred_svm_rbf=
rbf.predict(vectorizer.transform([translate_text]))
    if y_pred_svm_rbf == 'neutral':
        messagebox.showinfo(title='Results',
message='Hasil analisis dari sentimen komentar kamu
adalah cyberbullying netral')
    elif y_pred_svm_rbf == 'positive':
        messagebox.showinfo(title='Results',
message='Hasil analisis dari sentimen komentar kamu
adalah cyberbullying positif')
    else :
        messagebox.showinfo(title='Results',
message='Hasil analisis dari sentimen komentar kamu

```

```

message='Hasil analisis dari sentimen komentar kamu
adalah cyberbullying negatif')
    textcomment.delete(1.0, END)

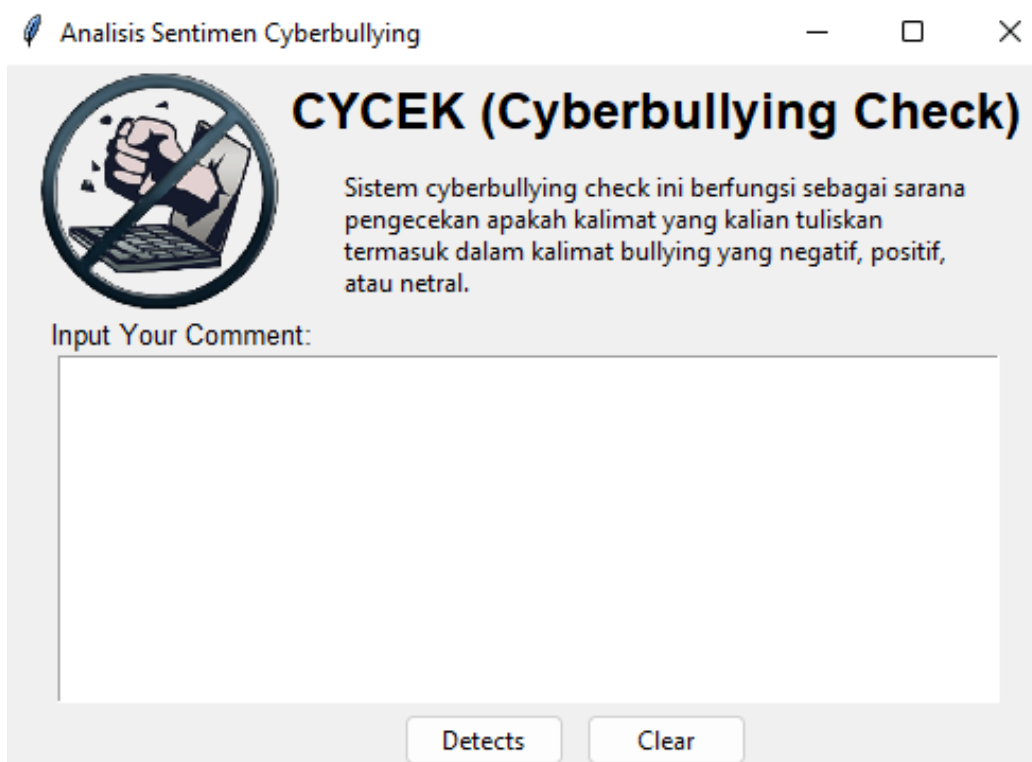
submitbutton = ttk.Button(frame_content, text='Detects',
command=submit).grid(row=2, column=0, padx=5, pady=5,
sticky='e')
clearbutton = ttk.Button(frame_content, text='Clear',
command=clear).grid(row=2, column=1, padx=5, pady=5,
sticky='w')

root.geometry('500x340')
mainloop()

```

*Gambar 8. Potongan script design UI*

Script diatas juga sudah termasuk dalam pengkoneksian antara database cyberbullying. UI juga didesain sederhana agar user dapat dengan mudah menggunakannya. Terdapat deskripsi singkat tentang apa itu CYCEK (Cyberbullying Check), kolom inputan kalimat user, option detect dan clear.

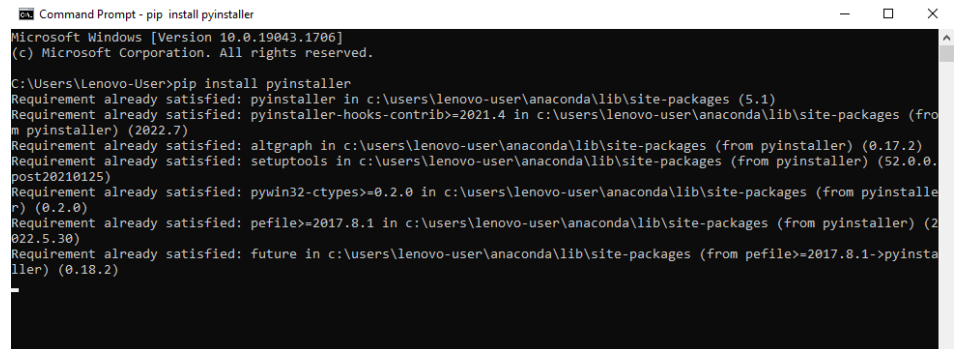


*Gambar 9. Tampilan UI Sistem*

#### 4.1.6 Merubah file python menjadi *executable file*

Tujuan dari dirubahnya menjadi executable file (.exe) adalah agar file mudah digunakan atau dioperasikan dengan mudah tanpa memerlukan aplikasi khusus. Adapun caranya adalah sebagai berikut :

A. Buka command prompt dan install pyinstaller menggunakan pip

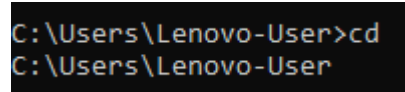


```
Microsoft Windows [Version 10.0.19043.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo-User>pip install pyinstaller
Requirement already satisfied: pyinstaller in c:\users\lenovo-user\anaconda\lib\site-packages (5.1)
Requirement already satisfied: pyinstaller-hooks-contrib>=2021.4 in c:\users\lenovo-user\anaconda\lib\site-packages (from pyinstaller) (2022.7)
Requirement already satisfied: altgraph in c:\users\lenovo-user\anaconda\lib\site-packages (from pyinstaller) (0.17.2)
Requirement already satisfied: setuptools in c:\users\lenovo-user\anaconda\lib\site-packages (from pyinstaller) (52.0.0.post20210125)
Requirement already satisfied: pywin32-ctypes>=0.2.0 in c:\users\lenovo-user\anaconda\lib\site-packages (from pyinstaller) (0.2.0)
Requirement already satisfied: pefile>=2017.8.1 in c:\users\lenovo-user\anaconda\lib\site-packages (from pyinstaller) (2022.5.30)
Requirement already satisfied: future in c:\users\lenovo-user\anaconda\lib\site-packages (from pefile>=2017.8.1->pyinstaller) (0.18.2)
```

Gambar 10. install pyinstaller menggunakan command prompt

B. Ketik cd untuk mencari lokasi file python yang akan dirubah



```
C:\Users\Lenovo-User>cd
C:\Users\Lenovo-User
```

Gambar 11. output cd

C. Letakkan file python ke lokasi yang muncul dari hasil cd

D. Ketikkan script berikut untuk mengubah file .py menjadi .exe

```
pyinstaller --onefile --add-data
"data_2_inggris_class.csv;data" --hidden-
import="sklearn.utils._cython_blas" --hidden-
import="sklearn.neighbors.typedefs" --hidden-
import="sklearn.neighbors.quad_tree" --hidden-
import="sklearn.tree._utils" --hidden-
import="sklearn.neighbors.typedefs" --hidden-
import="sklearn.utils._typedefs" --hidden-
import="sklearn.neighbors._partition_nodes" --
hidden-import="sklearn.utils._heap" --hidden-
import="sklearn.utils._sorting" --hidden-
import="sklearn.utils._vector_sentinel" --
icon=logo.ico cycek.py
```

Tabel 7. Script mengubah file menjadi exe

E. Tunggu proses instalasi hingga selesai

```
Microsoft Windows [Version 10.0.22000.739]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ACER>cd C:\Users\ACER\OneDrive\Documents\CYCEK\CYCEK

C:\Users\ACER\OneDrive\Documents\CYCEK\CYCEK>pyinstaller --onefile --add-data "data_2_inggris_class.csv;data" --hidden-i
import="sklearn.utils._cython_blas" --hidden-import="sklearn.neighbors.typedefs" --hidden-import="sklearn.neighbors.quad
tree" --hidden-import="sklearn.tree_utils" --hidden-import="sklearn.neighbors._typedefs" --hidden-import="sklearn.utils
._typedefs" --hidden-import="sklearn.neighbors._partition_nodes" --hidden-import="sklearn.utils._heap" --hidden-import="
sklearn.utils._sorting" --hidden-import="sklearn.utils._vector_sentinel" --icon=logo.ico cycek.py
205 INFO: PyInstaller: 5.1
206 INFO: Python: 3.9.7
226 INFO: Platform: Windows-10-10.0.22000-SP0
227 INFO: wrote C:\Users\ACER\OneDrive\Documents\CYCEK\CYCEK\cycek.spec
232 INFO: UPX is not available.
235 INFO: Extending PYTHONPATH with paths
['C:\Users\ACER\OneDrive\Documents\CYCEK\CYCEK']
1387 INFO: checking Analysis
2185 INFO: Appending 'datas' from .spec
2192 INFO: checking PYZ
2472 INFO: checking PKG
2809 INFO: Bootloader C:\Users\ACER\AppData\Local\Programs\Python\Python39\lib\site-packages\PyInstaller\bootloader\Wind
ows-64bit\run.exe
2810 INFO: checking EXE

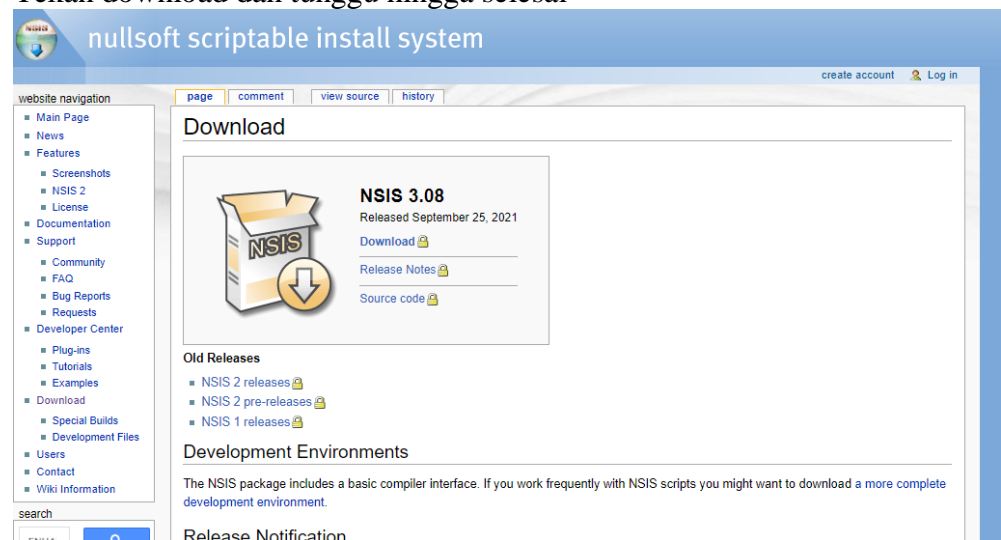
C:\Users\ACER\OneDrive\Documents\CYCEK\CYCEK>
```

Gambar 12. proses instalasi selesai

F. Jika proses instalasi selesai, file exe akan tersimpan di folder dist

#### 4.1.7 Merubah folder ke file exe instalasi menggunakan aplikasi NSIS

- A Ubah folder menjadi bentuk zip/rar
- B Download aplikasi NSIS melalui link berikut  
<https://nsis.sourceforge.io/Download>
- C Tekan download dan tunggu hingga selesai



Gambar 13. download NSIS

- D Install dan tunggu hingga prosesnya selesai
- E Buka aplikasi NSIS

## nullsoft scriptable install system 3.08

### Compiler

[Compile NSI scripts](#)  
[Installer based on .ZIP file](#)

### Documentation

[NSIS Users Manual](#)  
[Example scripts](#)  
[Modern UI 2](#)

### Plug-ins

[AdvSplash](#) - splash with fade in/out  
[Banner](#) - banner with custom text  
[BgImage](#) - background image  
[Dialer](#) - internet connection  
[Math](#) - math operations  
[nsDialogs](#) - custom wizard pages  
[nsExec](#) - launch command line tools  
[NSISdl](#) - download files  
[Splash](#) - splash screen  
[StartMenu](#) - Start Menu folder selection  
[System](#) - Windows API calls  
[VPatch](#) - update existing files

### Developer Center

Many more examples, tutorials, plug-ins and NSIS-related software are available at the online Developer Center.

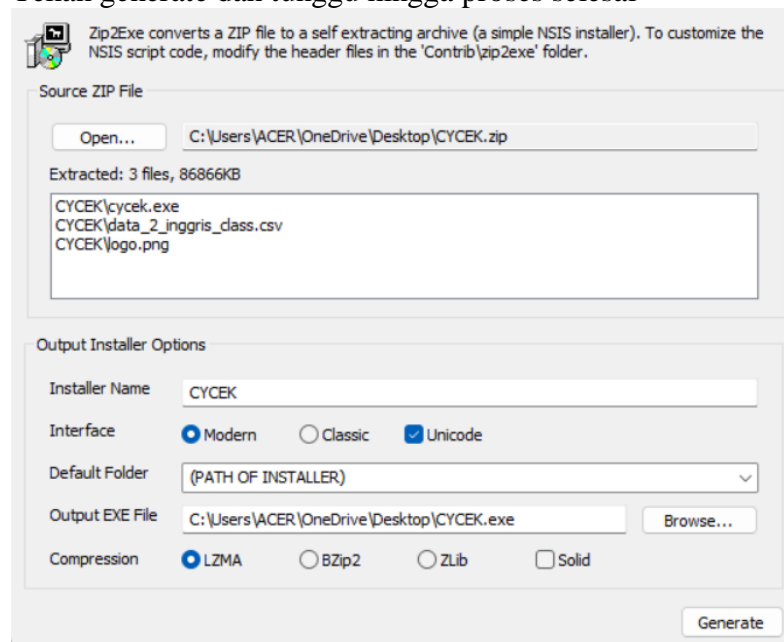
### Online Help

[Developer Center](#)  
[FAQ](#)  
[Forum](#)  
[Bug Tracker](#)  
[Stackoverflow](#)  
[Chat](#)

nsis.sourceforge.net

Gambar 14. aplikasi NSIS

- F Buka file zip yang akan diubah menjadi exe
- G Tentukan tempat penyimpanan file setelah proses menggunakan browse
- H Tekan generate dan tunggu hingga proses selesai



Gambar 15. proses convert

- I Ketika sudah selesai, file exe instalasi akan tersimpan sesuai dengan output exe file yang dipilih

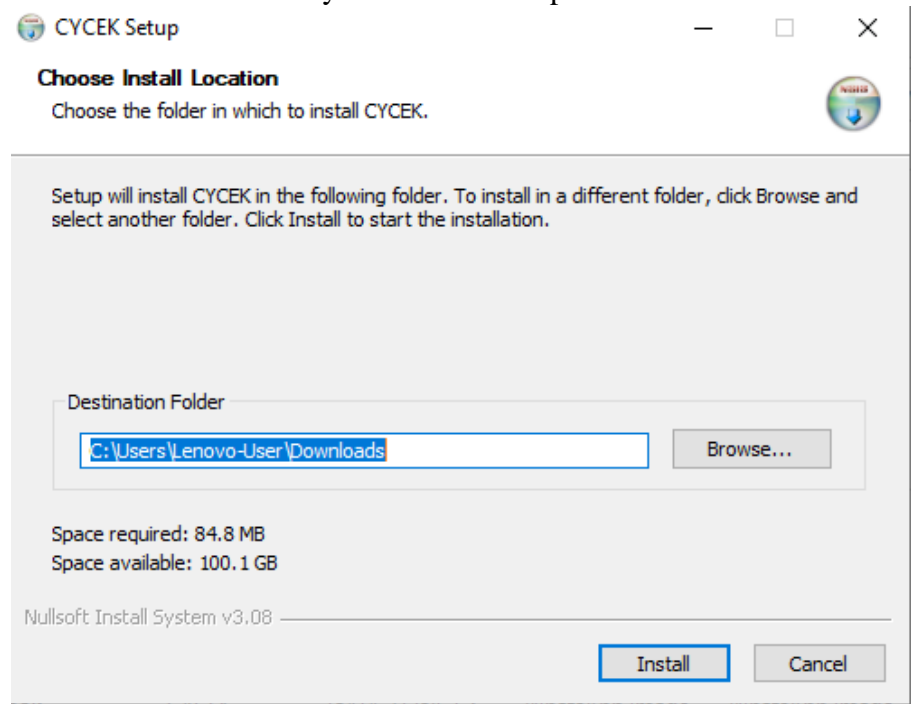
Name	Date modified	Type	Size
cycek	13/06/2022 15:29	File folder	
CYCEK	13/06/2022 15:41	Application	65.358 KB
cycek	13/06/2022 15:32	WinRAR ZIP archive	88.772 KB

Gambar 16. proses telah selesai

### 4.1.8 Cara menjalankan cycek.exe

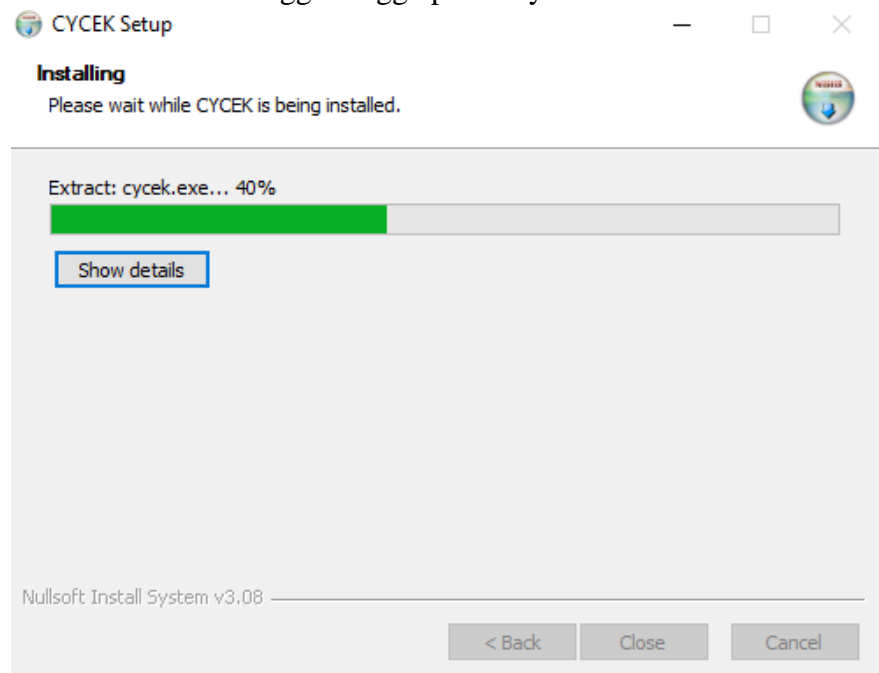
Setelah executable berhasil dibuat, untuk menjalankannya juga ada beberapa tahapan yang harus dilalui. Adapun tahapannya adalah sebagai berikut :

- A Buka file cycek.exe
- B Tentukan dimana nantinya file akan disimpan



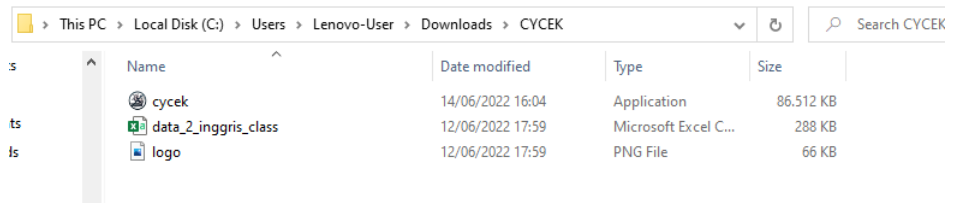
Gambar 17. menentukan lokasi file setelah di run

- C Tekan install dan tunggu hingga prosesnya selesai



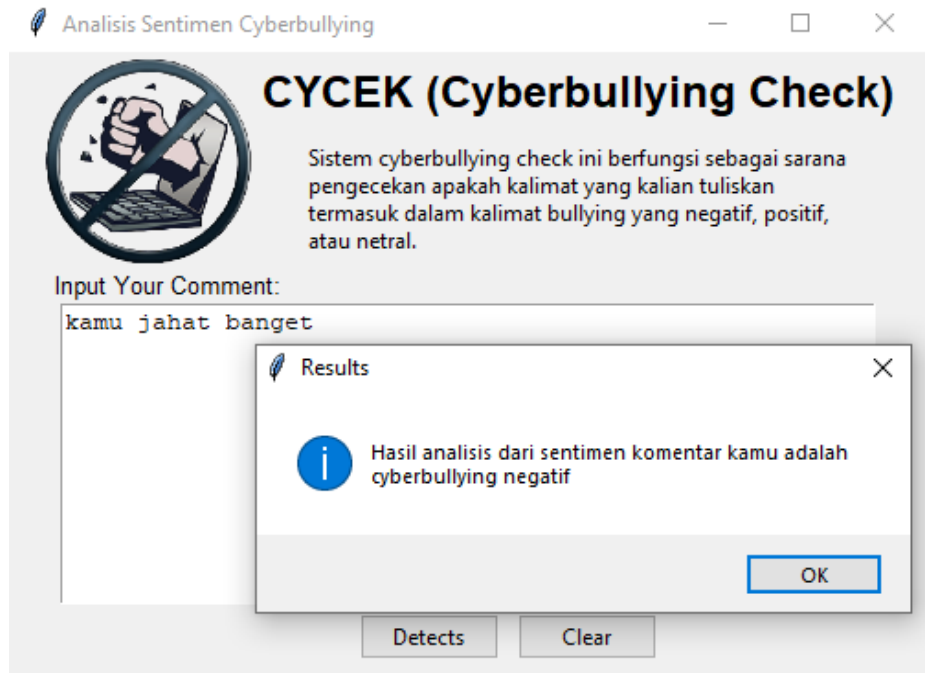
Gambar 18. proses install cycek.exe

- D Sebelum mencoba menjakankan, pastikan cycek.exe berada satu folder dengan file data\_2\_inggris\_class.csv dan juga logo.png



Gambar 19. peletakan satu folder

E Cycek.exe siap digunakan



Gambar 20. aplikasi cycek

## 4.2 Tingkat akurasi algoritma SVM

Dari hasil pengujian menggunakan kernel RBF, diperoleh tingkat akurasi sebesar 93,18%, *precision* sebesar 92,9%, dan *recal* sebesar 86,9% seperti yang terlihat pada gambar berikut,



---

```
Accuracy RBF : 93.181818181817 %
```

```
confusion matrix rbf :  
[[ 52  14   3]  
 [  2 394   4]  
 [  3  16 128]]
```

```
precision rbf : 0.9298913776404674
```

```
recall rbf : 0.8697904959085084
```

```
cross validation rbf : [0.93902439 0.87398374 0.8902439 0.8902439 0.8902439 0.92276423  
0.91056911 0.8902439 0.89430894 0.92276423]
```

*Gambar 21. Hasil akurasi algoritma SVM kernel RBF*

Sedangkan dari hasil pengujian menggunakan kernel linear, diperoleh tingkat akurasi sebesar 91,07%, *precision* sebesar 89,79% dan *recall* sebesar 84,31% seperti pada gambar berikut,

```
Accuracy Linear : 91.07142857142857 %
```

```
confusion matrix linear :  
[[ 52  16   1]  
 [  2 392   6]  
 [  8  22 117]]
```

```
precision linear : 0.8979619904976245
```

```
recall linear : 0.8431805185842453
```

```
cross validation linear : [0.93902439 0.88211382 0.87804878 0.8902439 0.88211382 0.89430894  
0.88617886 0.86585366 0.88211382 0.89837398]
```

*Gambar 22. Hasil akurasi algoritma SVM kernel linear*

### 4.3 Visualisasi Word Cloud

Gambar 8 menunjukkan visualisasi Word Cloud dari video berjudul “Video Amatir Detik-Detik Seorang Pelawak Ditangkap Polisi Karena Narkoba”.



Gambar 23. Hasil visualisasi word cloud

Deri hasil klasifikasi menunjukkan bahwa komentar terbanyak merupakan komentar netral. Memang komentar negatif cenderung sedikit, tetapi ketika kita melihat hasil visualisasinya, banyak kata-kata tidak senonoh yang mendominasi. Hal ini dikarenakan mayoritas komentar yang ada pada video berjudul “Video Amatir Detik-Detik Seorang Pelawak Ditangkap Polisi Karena Narkoba” banyak mengandung kata-kata tidak senonoh yang jika dianalisis tidak masuk dalam kalimat negatif, tetapi sebetulnya masuk dalam *cyberbullying* karena mengandung cacian.

## **BAB V: KESIMPULAN**

Berdasarkan hasil dari proses analisa yang telah dilakukan terhadap komentar sentimen “Video Amatir Detik-Detik Seorang Pelawak Ditangkap Polisi karena Narkoba” sebanyak 3.777 komentar. Komentar yang dapat diproses dalam klasifikasi hanyalah 3.076. Hal itu disebabkan banyak komentar yang berisikan kata-kata tidak berarti yang dihilangkan dalam text preprocessing. Model klasifikasi memprediksi terhadap sentimen negatif sebanyak 341 komentar (11,08%), sentimen positif sebanyak 737 (23,95%), sentimen netral sebanyak 1998 (64,95%).

Pada proses klasifikasi cenderung memiliki banyak komentar netral karena data komentar tersebut perlu diterjemahkan dahulu dalam bahasa inggris, hal ini tentu memberikan interpretasi yang berbeda. Dalam komentar juga banyak didapati sarkasme yang berarti komentar tersebut jika diklasifikasikan akan bernilai netral. Namun secara pemahaman manusia, komentar tersebut mengandung cyberbullying.

Dari hasil pengukuran performa model klasifikasi SVM, kernel RBF mendapatkan hasil keseluruhan yang paling baik dibanding dengan kernel Linear. Nilai akurasi, precision, dan recall yang dimiliki secara berturut-turut adalah 93,18%, 92,98% dan 86,97%.

## DAFTAR PUSTAKA

- Alkaff, Baim. 2016. Analisa Teks dengan Textblob. <http://baimalkaff.blogspot.com/2016/11/analisa-teks-dengan-textblob.html>. [Diakses pada 30 Maret 2022]
- Arsi, P., & Waluyo, R. (2021). Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM). Jurnal Teknologi Informasi dan Ilmu Komputer, 8(1), 147-156. doi:<http://dx.doi.org/10.25126/jtiik.0813944>
- Cahyono, A. S. (1). PENGARUH MEDIA SOSIAL TERHADAP PERUBAHAN SOSIAL MASYARAKAT DI INDONESIA. Publiciana, 9(1), 140-157. Retrieved from <https://journal.unita.ac.id/index.php/publiciana/article/view/79>
- Fitri, E. (2020). Analisis Sentimen Terhadap Aplikasi Ruanguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine. Jurnal Transformatika, 18(1), 71-80. doi:<http://dx.doi.org/10.26623/transformatika.v18i1.2317>
- hyunggg. 2021. Analisa Sentimen twiteter dengan algoritma Support Vector Machine (SVM) [Video]. YouTube. <https://www.youtube.com/watch?v=Sel8R6f0808>
- Hussein, Saddam. 2021. Support Vector Machine, Algoritma untuk Machine Learning. <https://geospasialis.com/support-vector-machine/>. [Diakses pada 30 Maret 2022]
- Katryn, Rianita Giovanni. Text Preprocessing: Tahap Awal dalam Natural Language Processing (NLP). 2020. <https://medium.com/mandiri-engineering/text-preprocessing-tahap-awal-dalam-natural-language-processing-nlp-bc5fbb6606a>. [Diakses pada 30 Maret 2022]
- Khaira, Johanda, Utomo, Suratno, (2020). Sentiment Analysis of Cyberbullying on Twitter Using SentiStrength. Indonesian Journal of Artificial Intelligence and Data Mining (IJAIMD). 3(1). 21-27. doi:<http://dx.doi.org/10.24014/ijaidm.v3i1.9145>
- Learning Orbis. 2020. Download or Scrape all YouTube Comments and Replies [Video]. YouTube. <https://youtu.be/uD58-EHwaeI>
- Luqyana, W. A., Cholissodin, I., Perdana, Rizal S. Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2(11), 4704-

4713. Retrieved from <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/3051/1195/>

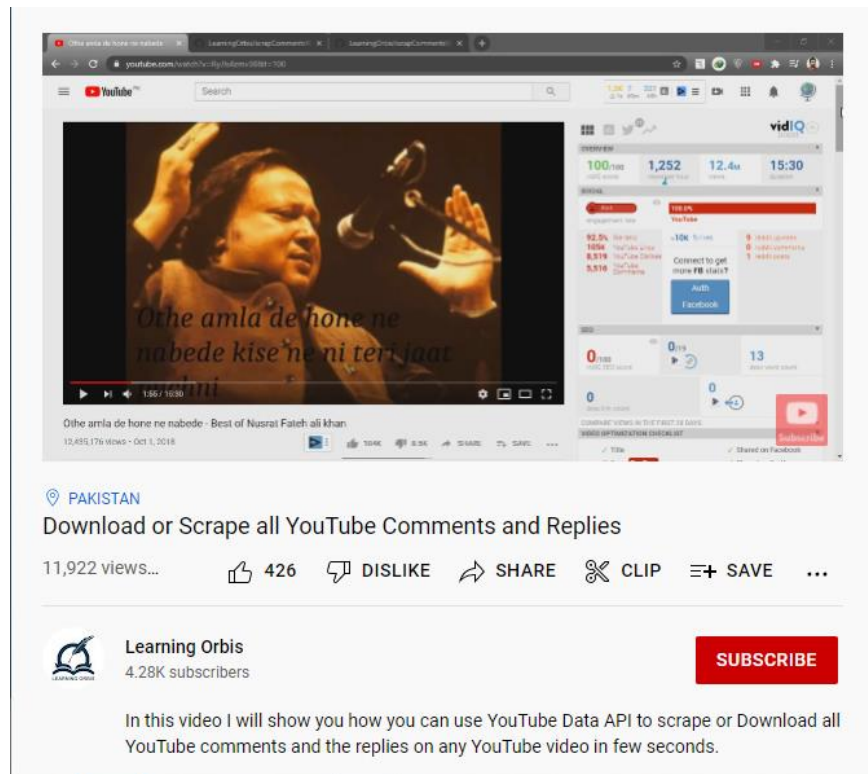
- Luqyana, W. A. (2018). Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine (Doctoral dissertation, Universitas Brawijaya).
- Marsinun, R., & Dody, R. (2020). Perilaku Cyberbullying Remaja di Media Sosial. *Analitika: Jurnal Magister Psikologi UMA*, 12 (2): 98 - 111. doi:<https://ojs.uma.ac.id/index.php/analitika/article/view/3704/3022>
- Samosir, Fransiska. (2019). Efektifitas Pemanfaatan Youtube Melalui Video Online Sebagai Media Pembelajaran Oleh Mahasiswa (Studi Di Fakultas Fisip Universitas Bengkulu). *Record and Library Journal*. 4. 81. 10.20473/rlj.V4-I2.2018.81-91.
- Saraswati, Rimirasih, (2020). ANALISIS SENTIMEN TERHADAP PELAYANAN KRL COMMUTERLINE BERDASARKAN DATA TWITTER MENGGUNAKAN ALGORITMA BERNOULLI NAIVE BAYES. *Jurnal Ilmiah Informatika Komputer*. 25(3), 225-238. doi: <https://doi.org/10.35760/ik.2020.v25i3.3256>
- Sari, F., & Wibowo, A. (2019). ANALISIS SENTIMEN PELANGGAN TOKO ONLINE JD.ID MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER BERBASIS KONVERSI IKON EMOSI. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 10(2), 681-686. doi:<https://doi.org/10.24176/simet.v10i2.3487>
- Subagyo, Subagyo and Ramadhani, Olivia and Mardiah, Hayyatul and Ariba, Rayi Arkan (2020) Pemanfaatan Data Media Sosial untuk Identifikasi Awal Karakter Produk. In: Seminar Nasional Teknik Industri (SENTI) UGM, 5 Oktober 2020, Yogyakarta.
- Tineges, Rian. 2021. Tahapan Text Preprocessing dalam Teknik Pengolahan Data. <https://dqqlab.id/tahapan-text-preprocessing-dalam-teknik-pengolahan-data>. [Diakses 25 Maret 2022].
- TNI Informatika'17. 2020. Analisis Sentimen menggunakan metode Support Vector Machine [Video]. YouTube. <https://www.youtube.com/watch?v=3DJ3mCIKNWA>

TNI Informatika'17. 2020. SENTIMENT ANALYSIS MENGGUNAKAN METODE  
SUPPORT VECTOR MACHINE(SVM) [Video]. YouTube.  
<https://www.youtube.com/watch?v=lW2fb9-WRLg>

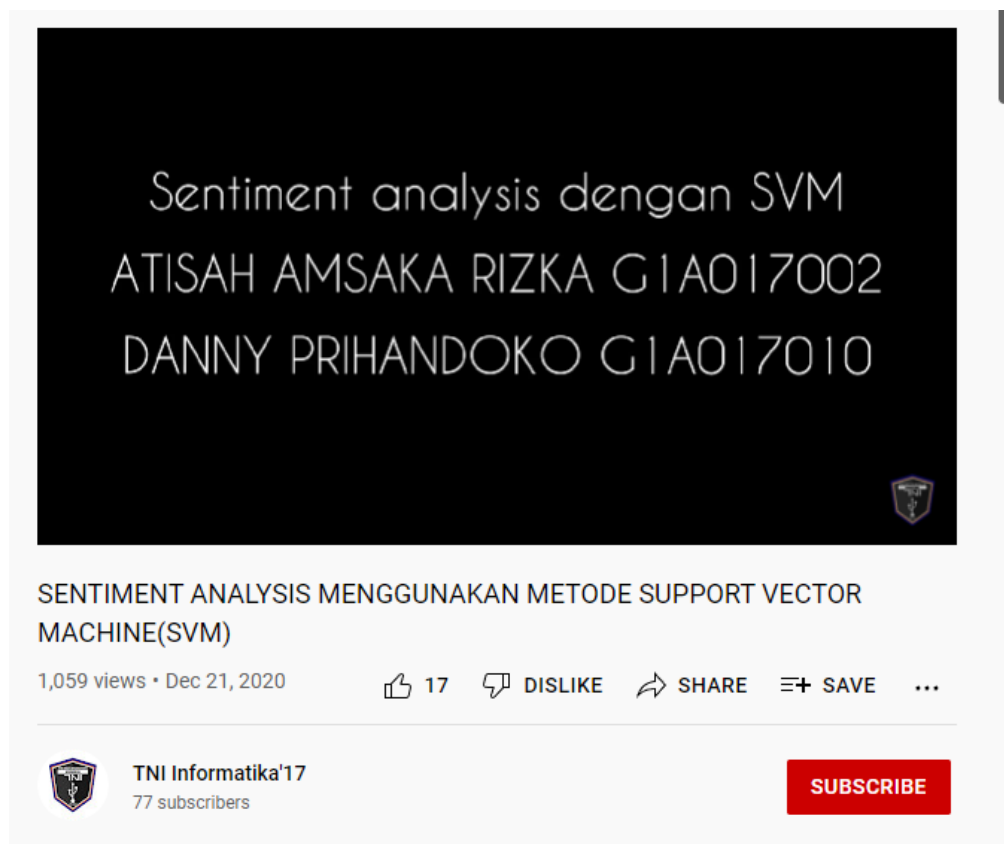
## LAMPIRAN



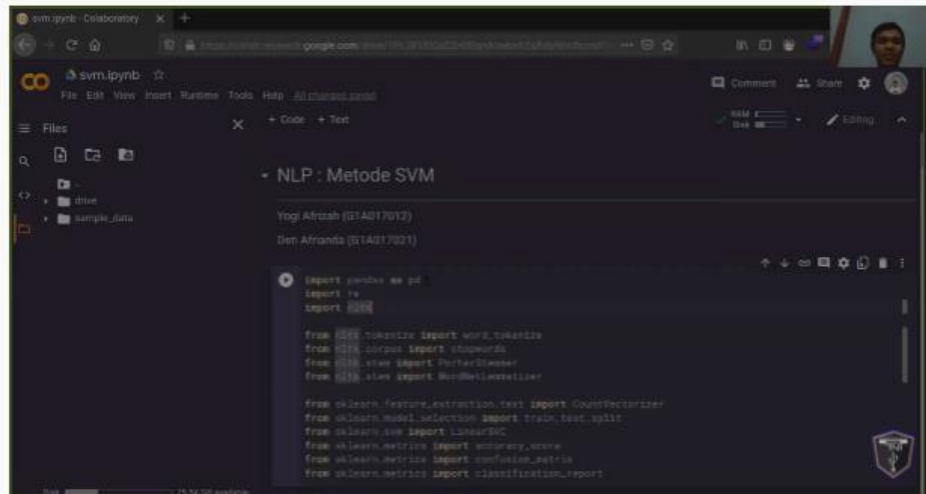
Gambar 24. Video yang di analisis



Gambar 25. Tutorial scraping dan download komentar video YouTube



Gambar 26. Tutorial coding 1



The screenshot shows a Jupyter Notebook interface with a dark theme. The title bar reads 'svm.ipynb - Colaboratory'. The left sidebar shows a file explorer with folders 'drive' and 'sample\_data'. The main area has a title 'NLP : Metode SVM' and two authors listed: 'Yogi Afrizah (G1A017012)' and 'Deri Afrinda (G1A017021)'. The code cell contains the following Python code:

```
import pandas as pd
import re
import nltk

from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer

from sklearn.feature_extraction.text import CountVecorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

## Analisis Sentimen menggunakan metode Support Vector Machine

2,515 views • Dec 29, 2020

48

DISLIKE

SHARE

SAVE

...



**TNI Informatika'17**

77 subscribers

SUBSCRIBE

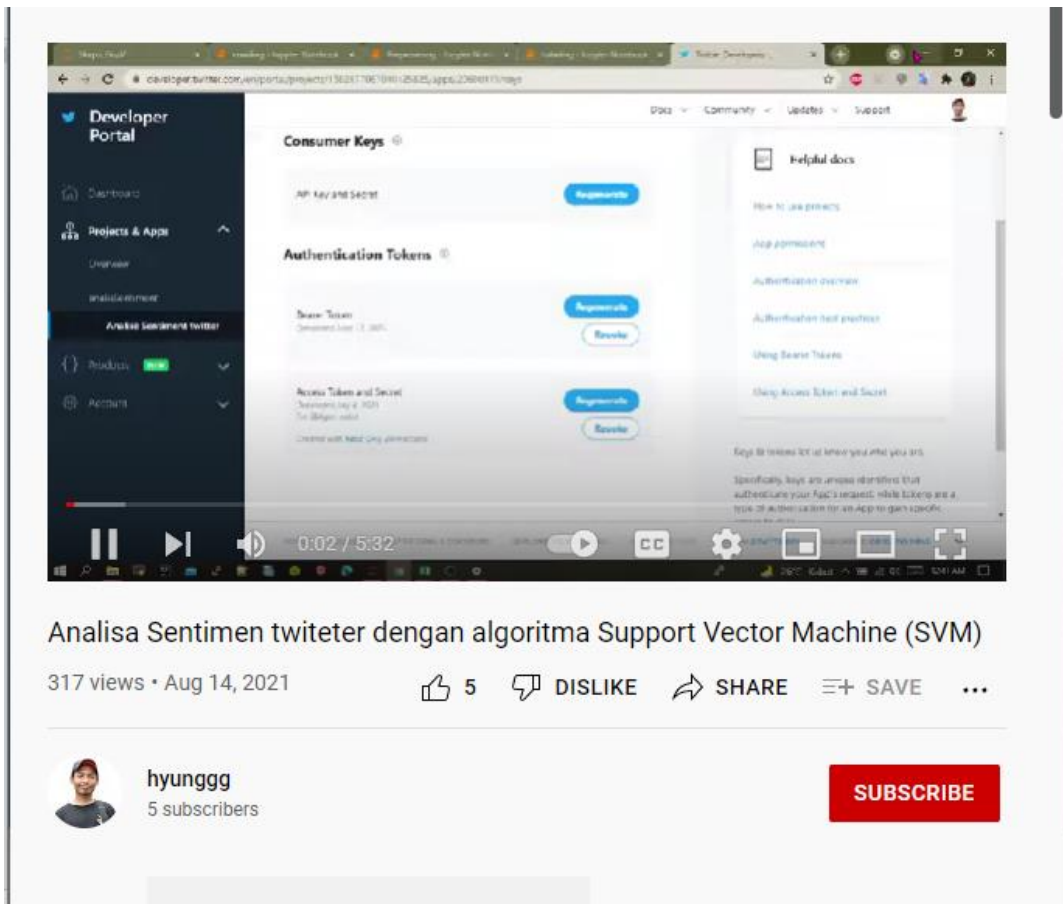
YOGI AFRIZAH (G1A017012)

DERI AFRIANDA (G1A017021)

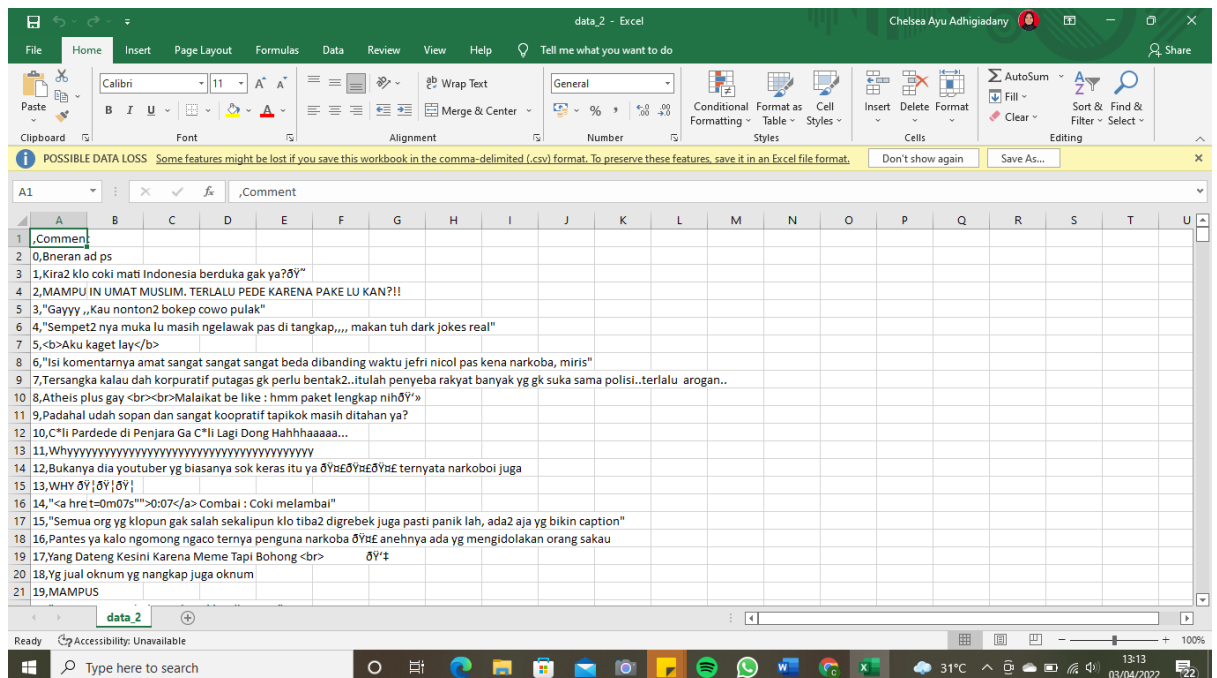
SHOW MORE

Gambar 27. Tutorial coding 2





Gambar 28. Tutorial coding 3



Gambar 29. dataset hasil download komentar YouTube