

**ANALISIS SENTIMEN KOMENTAR CYBERBULLYING  
PADA MEDIA SOSIAL INSTAGRAM MENGGUNAKAN  
METODE SUPPORT VECTOR MACHINE (SVM)**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Persyaratan Guna Meraih Gelar Sarjana  
Informatika Universitas Muhammadiyah Malang



Cita Tiara Hanni

201710370311167

**Bidang Minat**

Data Science

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH MALANG  
2021**

## LEMBAR PERSETUJUAN

### ANALISIS SENTIMEN KOMENTAR *CYBERBULLYING* PADA MEDIA SOSIAL INSTAGRAM MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE (SVM)*

#### TUGAS AKHIR

Sebagai Persyaratan Guna Meraih Gelar Sarjana Strata 1  
Informatika Universitas Muhammadiyah Malang

Menyetujui,  
Malang, 2021

Pembimbing I

Pembimbing II



Christian Sri K. A. S.Kom, M.Kom

NIP. 180327021991



Didih Rizki C. S.Kom, M.Kom

NIDN. 0702109201

## LEMBAR PERSETUJUAN

# ANALISIS SENTIMEN KOMENTAR *CYBERBULLYING* PADA MEDIA SOSIAL INSTAGRAM MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE (SVM)*

## TUGAS AKHIR

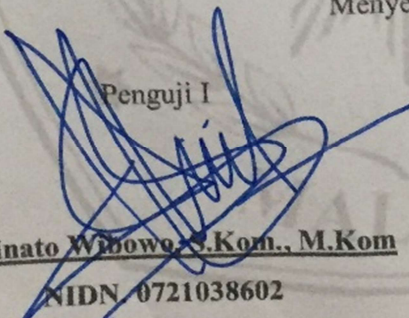
Sebagai Persyaratan Guna Meraih Gelar Sarjana Strata 1  
Informatika Universitas Muhammadiyah Malang

Disusun Oleh:  
**Cita Tiara Hanni**  
**201710370311167**

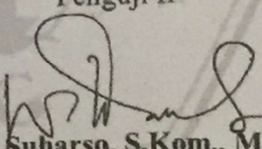
Tugas Akhir ini telah diuji dan dinyatakan lulus melalui siding majelis penguji  
pada tanggal 19 Oktober 2021

Menyetujui,

Penguji I

  
**Hardinato Wibowo, S.Kom., M.Kom**  
**NIDN. 0721038602**

Penguji II

  
**Wildan Suharso, S.Kom., M.Kom**  
**NIDN. 0730038405**

Mengetahui,

Ketua Jurusan Informatika

  
**Hi. Giha Indah Marthasari ST., M.Kom**  
**NIP. 108.06110442**

## LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini:

**NAMA : CITA TIARA HANNI**

**NIM : 201710370311167**

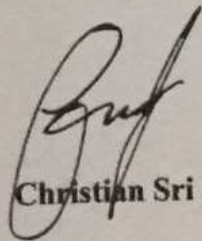
**FAK./JUR. : TEKNIK/INFORMATIKA**

Dengan ini saya menyatakan bahwa Tugas Akhir dengan judul **"ANALISIS SENTIMEN KOMENTAR *CYBERBULLYING* PADA MEDIA SOSIAL INSTAGRAM MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE (SVM)*"** beserta seluruh isinya adalah karya saya sendiri dan bukan merupakan karya tulis orang lain, baik sebagian maupun seluruhnya, kecuali dalam bentuk kutipan yang telah disebutkan sumbernya.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya. Apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini maka saya siap menanggung segala bentuk resiko/sanksi yang berlaku.

Mengetahui,

Dosen Pembimbing



**Christian Sri K. A. S. Kom, M. kom**

Malang,

Yang Membuat Pernyataan



**Cita Tiara Hanni**

## KATA PENGANTAR

Dengan memanjatkan puji syukur kehadiran Allah SWT. Atas limpahan rahmat dan hidayah-NYA sehingga peneliti dapat menyelesaikan tugas akhir yang berjudul:

**“ANALISIS SENTIMEN KOMENTAR *CYBERBULLYING* PADA  
MEDIA SOSIAL INSTAGRAM MENGGUNAKAN METODE *SUPPORT  
VECTOR MACHINE (SVM)*”**

Di dalam tulisan ini disajikan pokok-pokok bahasan yang meliputi latar belakang, metode penelitian, serta hasil dan pembahasan yang diperoleh dari penelitian ini dan telah disimpulkan berdasarkan hasil yang didapatkan oleh peneliti.

Peneliti menyadari sepenuhnya bahwa dalam penulisan tugas akhir ini masih banyak kekurangan dan keterbatasan. Oleh karena itu peneliti mengharapkan saran yang membangun agar tulisan ini bermanfaat bagi perkembangan ilmu pengetahuan.

Malang, 28 Mei 2021

Cita Tiara Hanni

## DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PERSETUJUAN.....	iv
LEMBAR PERNYATAAN.....	iii
ABSTRAK .....	iv
<i>ABSTRACT</i> .....	v
LEMBAR PERSEMBAHAN .....	vi
KATA PENGANTAR.....	viii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xii
DAFTAR TABEL .....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian .....	3
1.4 Batasan Masalah.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Penelitian Terdahulu.....	5
2.2 Text Mining.....	6
2.3 Analisis Sentimen .....	7
2.4 Cyberbullying .....	7
2.5 Media Sosial .....	7
2.6 Instagram .....	8
2.7 Teks Pre-processing .....	9
2.7.1 Case Folding.....	9
2.7.2 Cleaning.....	9
2.7.3 Normalisasi .....	10
2.7.4 Stopword Removal.....	10
2.7.5 Stemming .....	10
2.7.6 Tokenizing .....	10
2.8 TF-IDF .....	10

2.9	Support Vector Machine.....	10
2.10	Pengukuran Performa .....	12
2.10.1	Precision.....	12
2.10.2	Recall.....	12
2.10.3	Accuracy .....	12
<b>BAB III METODOLOGI PENELITIAN .....</b>		<b>14</b>
3.1	Rancangan Penelitian.....	14
3.2	Pengumpulan Data .....	15
3.3	Pre-processing.....	18
3.3.1	Case Folding .....	18
3.3.2	Cleaning .....	18
3.3.3	Normalisasi .....	19
3.3.4	Stopword Removal.....	20
3.3.5	Stemming .....	20
3.3.6	Tokenizing .....	21
3.4	Pembobotan TF-IDF .....	21
3.5	Split Data .....	25
3.6	Pemodelan SVM .....	25
3.7	Evaluasi Model .....	26
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>		<b>28</b>
4.1	Implementasi Sistem .....	28
4.1.1	Hardware.....	28
4.1.2	Software .....	28
4.2	Library .....	29
4.3	Dataset .....	29
4.4	Pre-Processing .....	30
4.4.1	Cleaning .....	31
4.4.2	Case Folding.....	31
4.4.3	Tokenizing .....	31
4.4.4	Normalisasi .....	32
4.4.5	Stopword Removal.....	32
4.4.6	Steming .....	33
4.5	Word Cloud.....	33

4.6	TF-IDF.....	35
4.7	Scalling and Transform .....	35
4.8	Pemodelan Support Vector Machine.....	35
4.8.1	Tunning Hyperparameter .....	35
4.8.2	Gridsearch CV .....	36
4.8	Classification Report.....	37
4.9	Confusion Matrix .....	38
4.10	Evaluasi Model .....	41
BAB V KESIMPULAN.....		43
DAFTAR PUSTAKA .....		44



## DAFTAR GAMBAR

<b>Gambar 2.1</b> Tahapan <i>pre-pocessing</i> .....	9
<b>Gambar 3.1</b> Alur penelitian .....	14
<b>Gambar 3.2</b> Perhitungan TF-IDF.....	22
<b>Gambar 3.3</b> <i>Split data Support Vector Machine</i> .....	25
<b>Gambar 3.4</b> Tahapan metode SVM .....	26
<b>Gambar 4.1</b> <i>Source code import library</i> .....	29
<b>Gambar 4.2</b> <i>Syntax import Google Drive pada Google Colab</i> .....	30
<b>Gambar 4.3</b> Dataset komentar <i>cyberbullying</i> di Instagram .....	30
<b>Gambar 4.4</b> <i>Syntax tahapan cleaning</i> .....	31
<b>Gambar 4.5</b> <i>Syntax tahapan case folding</i> .....	31
<b>Gambar 4.6</b> <i>Syntax tahapan tokenizing</i> .....	32
<b>Gambar 4.7</b> <i>Syntax tahapan normalisasi</i> .....	32
<b>Gambar 4.8</b> <i>Syntax tahapan stopword removal</i> .....	32
<b>Gambar 4.9</b> <i>Syntax tahapan stemming</i> .....	32
<b>Gambar 4.10</b> <i>Source code word cloud non-bullying</i> .....	33
<b>Gambar 4.11</b> <i>Source code word cloud bullying</i> .....	34
<b>Gambar 4.12</b> <i>Word cloud non-bullying</i> .....	34
<b>Gambar 4.13</b> <i>Word cloud bullying</i> .....	34
<b>Gambar 4.14</b> <i>Source code TF-IDF</i> .....	35
<b>Gambar 4.15</b> <i>Syntax scalling and transform</i> .....	35
<b>Gambar 4.16</b> <i>Syntax tunning hyperparameter</i> .....	36
<b>Gambar 4.17</b> <i>Source code pemodelan SVM</i> .....	36
<b>Gambar 4.18</b> <i>Output CV result</i> .....	37
<b>Gambar 4.19</b> <i>Syntax classification report</i> .....	38
<b>Gambar 4.20</b> <i>Output classification report</i> .....	38
<b>Gambar 4.21</b> <i>Syntax confusion matrix</i> .....	38
<b>Gambar 4.22</b> <i>Confusion matrix split data 90:10</i> .....	39
<b>Gambar 4.23</b> <i>Confusion matrix split data 80:20</i> .....	39
<b>Gambar 4.24</b> <i>Confusion matrix split data 70:30</i> .....	40
<b>Gambar 4.25</b> <i>Confusion matrix split data 60:40</i> .....	41

## DAFTAR TABEL

Tabel 2.1 Penelitian terdahulu.....	5
Tabel 2.2 <i>Confusion matrix</i> .....	12
Tabel 3.1 <i>Dataset</i> .....	15
Tabel 3.2 Proses <i>case folding</i> .....	18
Tabel 3.3 Proses <i>cleaning</i> .....	19
Tabel 3.4 Proses normalisasi .....	19
Tabel 3.5 Proses <i>stopword removal</i> .....	20
Tabel 3.6 Proses <i>stemming</i> .....	21
Tabel 3.7 Proses <i>tokenizing</i> .....	21
Tabel 3.8 Komentar Instagram.....	22
Tabel 3.9 <i>Inverted index</i> .....	23
Tabel 3.10 Perhitungan TF-IDF .....	24
Tabel 3.11 <i>Confusion matrix</i> .....	27
Tabel 4.1 <i>Hardware</i> .....	28
Tabel 4.2 <i>Software</i> .....	28
Tabel 4.3 Perbandingan <i>precision, recall, f1-measure</i> , dan <i>accuracy</i> .....	41

## DAFTAR PUSTAKA

- [1] H. M. A. I. Amali, "Classification of Cyberbullying Sinhala Language Comments on Social Media," pp. 266–271, 2020.
- [2] D. L. Espelage and J. S. Hong, "Cyberbullying Prevention and Intervention Efforts: Current Knowledge and Future Directions," *Canadian Journal of Psychiatry*, vol. 62, no. 6, pp. 374–380, 2017, doi: 10.1177/0706743716684793.
- [3] T. Febriana and A. Budiarto, "Twitter Dataset for Hate Speech and Cyberbullying Detection in Indonesian Language," *Proceedings of 2019 International Conference on Information Management and Technology, ICIMTech 2019*, vol. 1, no. August, pp. 379–382, 2019, doi: 10.1109/ICIMTech.2019.8843722.
- [4] R. R. Dalvi, S. Baliram Chavan, and A. Halbe, "Detecting A Twitter Cyberbullying Using Machine Learning," *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, no. Iciccs, pp. 297–301, 2020, doi: 10.1109/ICICCS48265.2020.9120893.
- [5] D. Chatzakou *et al.*, "Detecting cyberbullying and cyberaggression in social media," *arXiv*, vol. 13, no. 3, 2019.
- [6] M. A. Al-Garadi *et al.*, "Predicting Cyberbullying on Social Media in the Big Data Era Using Machine Learning Algorithms: Review of Literature and Open Challenges," *IEEE Access*, vol. 7, no. c, pp. 70701–70718, 2019, doi: 10.1109/ACCESS.2019.2918354.
- [7] R. M. Candra and A. Nanda Rozana, "Klasifikasi Komentar Bullying pada Instagram Menggunakan Metode K-Nearest Neighbor," *IT Journal Research and Development*, vol. 5, no. 1, pp. 45–52, 2020, doi: 10.25299/itjrd.2020.vol5(1).4962.
- [8] N. M. G. D. Purnamasari, M. A. Fauzi, Indriarti, and L. S. Dewi, "Identifikasi Tweet Cyberbullying pada Aplikasi Twitter menggunakan Metode Support Vector Machine ( SVM ) dan Information Gain ( IG ) sebagai Seleksi Fitur," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 11, pp. 5326–5332, 2018.

- [9] W. A. Luqyana, I. Cholissodin, and R. S. Perdana, "Analisis Sentimen Cyberbullying Pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, vol. 2, no. 11, pp. 4704–4713, 2018.
- [10] Z. Halim, M. Waqar, and M. Tahir, "Knowledge-Based Systems A machine learning-based investigation utilizing the in-text features for the identification of dominant emotion in an email," *Knowledge-Based Systems*, vol. 208, p. 106443, 2020, doi: 10.1016/j.knosys.2020.106443.
- [11] A. I. Kadhim, "An Evaluation of Preprocessing Techniques for Text Classification," *International Journal of Computer Science and Information Security*, vol. 16, no. 6, pp. 22–32, 2018.
- [12] Y. Chen, "Mining of instant messaging data in the Internet of Things based on support vector machine," *Computer Communications*, vol. 154, no. March, pp. 278–287, 2020, doi: 10.1016/j.comcom.2020.02.080.
- [13] Y. Win, "Classification using Support Vector Machine to Detect Cyberbullying in Social Media for Myanmar Language," *2019 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, pp. 122–125, 2019.
- [14] F. A. Yusup, M. A. Bijaksana, and A. F. Huda, "ScienceDirect Narrator ' s Name Recognition with Support Vector Machine for Narrator ' s Name Recognition with Support Vector Machine for Indexing Indonesian Hadith translations Indexing Indonesian Hadith translations," *Procedia Computer Science*, vol. 157, pp. 191–198, 2019, doi: 10.1016/j.procs.2019.08.157.
- [15] S. E. Saad and J. Yang, "Twitter Sentiment Analysis Based on Ordinal Regression," *IEEE Access*, vol. 7, pp. 163677–163685, 2019, doi: 10.1109/ACCESS.2019.2952127.



**UNIVERSITAS MUHAMMADIYAH MALANG**  
**FAKULTAS TEKNIK**  
**PROGRAM STUDI TEKNIK INFORMATIKA**  
Jl. Raya Tlogomas 246 Malang 65144 Telp. 0341 - 464318 Ext. 247, Fax. 0341 - 460782

**FORM CEK PLAGIARISME LAPORAN TUGAS AKHIR**

Nama Mahasiswa : Cita Tiara Hanni  
NIM : 201710370311167  
Judul TA : ANALISIS SENTIMEN KOMENTAR CYBERBULLYING PADA MEDIA  
SOSIAL INSTAGRAM MENGGUNAKAN METODE SUPPORT VECTOR  
MACHINE (SVM)

Hasil Cek Plagiarisme dengan Turnitin

No.	Komponen Pengecekan	Nilai Maksimal Plagiarisme (%)	Hasil Cek Plagiarisme (%) *
1.	Bab 1 – Pendahuluan	10 %	6%
2.	Bab 2 – Daftar Pustaka	25 %	19%
3.	Bab 3 – Analisis dan Perancangan	25 %	11%
4.	Bab 4 – Implementasi dan Pengujian	15 %	13%
5.	Bab 5 – Kesimpulan dan Saran	5 %	0%
6.	Makalah Tugas Akhir	20%	19%

Mengetahui,

Dosen Pembimbing

(Christian Sri Kusuma Aditya, S.Kom., M.Kom.)

\*) Hasil cek plagiarism bisa diisikkan oleh salah satu pembimbing

# BAB I

## PENDAHULUAN

Pada bab ini berisikan alasan diambilnya topik Tugas Akhir ini, apa saja rumusan permasalahannya yang diambil, tujuan penelitian, dan batasan-batasan masalah di penulisan Tugas Akhir ini.

### 1.1 Latar Belakang

*Smartphone* di era sekarang sudah menjadi kebutuhan wajib, dengan adanya paket data yang terjangkau, situs media sosial menjadi sangat populer saat ini. Penggunaan teknologi telah berkembang drastis, demikian juga kemampuan seseorang untuk mem-*bully* berkembang selama beberapa tahun terakhir. Sasaran *bullying* berpindah dari batasan fisik ke platform online [1]. *Cyberbullying* diartikan sebagai tindakan agresif dan disengaja untuk merugikan satu orang dimana sekelompok orang atau individu dengan memakai perangkat elektronik yang dilakukan berulang atau setiap saat ke korban yang tidak bisa menolong diri sendiri [2]. *Bullying* termasuk kejahatan yang tidak bisa dianggap remeh karena dapat berdampak pada fisik, psikis, hingga prestasi akademik korban. Sebagian besar korban akan memiliki tingkat *anxiety*, depresi, dan kepercayaan diri yang rendah. *Cyberbullying* dapat dilakukan melalui berbagai media seperti *text messages*, gambar video, telepon, *email* (surat elektronik), *chat rooms*, *instant messaging* (IM), berbagai situs media sosial, dan *website*. Media yang tercatat paling sering terjadi *cyberbullying* adalah media sosial [3]. Media sosial menjadi platform dimana banyak anak muda di *bully*. Lebih dari separuh pengguna muda media sosial di seluruh dunia terkena pelecehan digital dalam jangka waktu yang panjang. Penyalahgunaan teknologi media sosial telah memperkenalkan bentuk baru agresi dan kekerasan yang terjadi secara eksklusif secara online. Bertahun-tahun, media sosial telah berkembang besar dalam hal meningkatnya jumlah situs dan juga pengguna baru. Saat situs jejaring media sosial meningkat, *cyberbullying* juga semakin meningkat dari hari ke hari [4] [5] [1] [6].

Pada penelitian yang dilakukan oleh R. M. Candra, dkk pada tahun 2020 yang berjudul “*Classification of Instagram Bullying Comments Using the K-Nearest Neighbour Methods*” [7], *dataset* untuk penelitian ini berjumlah 1.000 data

yang diperoleh dari komentar yang terdapat pada *account* Instagram artis maupun selebgram Indonesia yang punya *followers* diatas 500 ribu, dimana 500 data dikategorikan *bullying* dan 500 *non-bully*. Pada tahap pengujian digunakan 5 nilai *k* berbeda (7, 9, 11, 13, 15) dan menggunakan 3 perbandingan (0.7:0.3, 0.8:0.2, dan 0.9:0.1) yang memakai *confusion matrix*, yang tiap-tiap *fold* yang dipakai 4, 5 & 10 hingga menjadikan total *fold* sebesar 95. Model yang digunakan pada penelitian ini ialah *K-Nearest Neighbour*. Pengujian pertama dengan perbandingan 70:30 (*test data*: 300, *train data*: 700) dibagi menjadi 4-fold mendapatkan akurasi tertinggi pada fold ke-3 sebesar 56%, pengujian kedua dengan perbandingan 80:20 (*test data*: 200, *train data*: 800) dibagi menjadi 5-fold mendapatkan akurasi tertinggi pada fold ke-5 sebesar 61,5%. Dan pengujian terakhir dengan perbandingan 90:10 (*test data*: 100, *train data*: 900) dibagi menjadi 10-fold mendapatkan *score accuracy* paling bagus di *fold* ke-6 sebesar 77%.

Pada penelitian sebelumnya oleh N. M. G. D. Purnamasari, dkk pada tahun 2018 dengan judul “*Identification of Cyberbullying Tweets on Twitter Application Using Support Vector Machine (SVM) Methods and Information Gain (IG) as feature selection*” [8], data pada penelitian ini diambil menggunakan R-Studio dari API Twitter dimana menyebut Wakil Ketua DPR RI, Fadli Zon dalam *tweet*-nya. Model yang diaplikasikan pada penelitian ini yaitu *Support Vector Machine (SVM)* menggunakan *Information Gain (IG)* sebagai seleksi fitur. Hasil yang diperoleh dengan penggunaan metode *Support Vector Machine (SVM)* yaitu akurasi sebesar 75% dengan skor presisi 70.27%, *recall* 86.66% dan *f1-score* 77.61%. Sedangkan skor uji *threshold feature selection information gain (IG)* skor terbagus diperoleh pada skor *threshold* 90% dimana skor akurasi sebesar 76,66%, skor presisi 72,22%, skor *recall* 86,66%, dan skor *f1-score* 78,78%.

Selanjutnya pada penelitian oleh R. R. Dalvi, dkk pada tahun 2020 dengan judul “*Detecting A Twitter Cyberbullying Using Machine Learning*” [4], data pada penelitian ini diambil dari Twitter API. Model yang dipakai pada penelitian ini ada 2, yaitu: *Naïve Bayes* dan *Support Vector Machine (SVM)*. Pada penelitian ini, skor *accuracy* yang didapat dari pengujian memakai metode *Support Vector Machine (SVM)* sebesar 71.25%, dan nilai *accuracy* menggunakan metode *Naïve Bayes* sebesar 52.70%.

Dari berbagai penelitian yang membahas tentang *cyberbullying* di Instagram ataupun Twitter dengan metode yang berbeda menghasilkan nilai performa yang beragam. Pada penelitian ini, untuk analisis sentiment *cyberbullying* di Instagram, metode yang diusulkan adalah *Support Vector Machine* (SVM) sehingga diharapkan mendapatkan nilai akurasi yang lebih bagus daripada penelitian sebelum-sebelumnya.

## **1.2 Rumusan Masalah**

Dari latar belakang yang disebutkan, maka didapatkan rumusan masalah untuk Tugas Akhir ini:

1. Bagaimana menerapkan metode *Support Vector Machine* (SVM) dalam meningkatkan akurasi pada analisis sentiment *cyberbullying* pada media sosial Instagram?
2. Bagaimana skor akurasi yang didapatkan dengan mengaplikasikan metode *Support Vector Machine* dalam analisis sentiment *cyberbullying* pada media sosial Instagram?

## **1.3 Tujuan Penelitian**

Tujuan pada pembuatan tugas akhir ini ialah guna mendapatkan hasil akurasi yang lebih baik dibandingkan penelitian sebelum-sebelumnya dengan menerapkan metode *Support Vector Machine* dengan algoritma *Linier Regression* dalam melakukan analisis sentiment *cyberbullying* pada media sosial Instagram.

## **1.4 Batasan Masalah**

Batasan masalah berisi apa saja yang diperlukan untuk penelitian, seperti jumlah data, model yang diaplikasikan, dll. Berikut batasan masalah dalam penulisan Tugas Akhir ini:

1. *Support Vector Machine* merupakan metode yang di aplikasikan pada penelitian Tugas Akhir ini.
2. *Dataset* pada penelitian Tugas Akhir dikumpulkan secara manual yang diperoleh dari komentar pada akun Instagram artist/selebgram yang

dituju sebagai objek. Terdapat 2 *class* yaitu: *bullying* dan *non-bullying* dimana pelabelan dilakukan secara manual dibantu oleh teman.

3. Total data pada penelitian ini sebanyak 650. Dengan pembagian 350 data termasuk kedalam *class bullying* dan 350 data lainnya masuk kedalam *class non-bullying*.



## BAB II

### TINJAUAN PUSTAKA

Pada bab ini berisikan penelitian terdahulu, uraian konsep *basic* beserta teori-teori yang bersangkutan pada topik penelitian Tugas Akhir ini dengan judul “Analisis Sentimen Komentar *Cyberbullying* pada Media Sosial Instagram Menggunakan Metode *Support Vector Machine* (SVM)”

#### 2.1 Penelitian Terdahulu

Berikut terdapat beberapa contoh penelitian yang sudah dilakukan sebelumnya dan digunakan sebagai patokan untuk menjalani penelitian ini:

**Tabel 2.1** Penelitian terdahulu

No	Penulis (Tahun)	Judul	<i>Dataset</i>	<i>Method</i>	Skor Akurasi
1	R. M. Candra, dkk. (2020)	Klasifikasi Komentar <i>Bullying</i> pada Instagram Menggunakan Metode <i>K-Nearest Neighbour</i>	1.000 Data	<i>K-Nearest Neighbour</i>	77%
2	N. M. G. D. Purnamasari, dkk. (2018)	Identifikasi <i>Tweet Cyberbullying</i> pada Aplikasi Twitter Menggunakan Metode <i>Support Vector Machine</i> (SVM) dan <i>Information Gain</i> (IG) sebagai Seleksi Fitur	300 <i>Tweets</i>	<i>Support Vector Machine</i> (SVM) memakai fitur seleksi <i>Information Gain</i> (IG)	76,66%

3	R. R. Dalvi, dkk. (2018)	<i>Detecting A Twitter Cyberbullying Using Machine Learning</i>	Twitter API	<i>Support Vector Machine dan Naïve Bayes</i>	71.25% (SVM), 52.70% (Naïve Bayes)
---	-----------------------------	---	-------------	---	--

## 2.2 Text Mining

*Text mining* yakni proses eksplorasi dan analisis sejumlah besar data teks (*tweet* pada Twitter, komentar pada Instagram, chat di WhatsApp, artikel pada media online) tidak terstruktur yang pengerjaannya menggunakan bantuan perangkat lunak yang bisa mengidentifikasi konsep, kata kunci, dan atribut lainnya pada data. Target dari *text mining* ialah memperoleh informasi yang bermanfaat dari kumpulan dokumen. Adapun tugas pokok dari *text mining* yaitu: kategorisasi teks (*text categorization*) dan pengelompokan teks (*text clustering*). Penggunaan *text mining* dilakukan untuk *cluster*, *classification*, *information retrieval*, dan *information extraction* [9].

Tahapan dalam pengerjaan *text mining* ada beberapa cara, yaitu:

1. *Knowledge Discovery Goal*,
2. *Data Preparation*,
3. *Data Pre-processing*,
4. *Data Modelling*,
5. *Evaluation*,
6. *Result*.

Manfaat dari *text mining* misalnya dalam dunia kesehatan dapat membantu mendiagnosis suatu penyakit dan keadaan pasien berdasarkan gejala yang mereka sebutkan. Dalam dunia bisnis, *text mining* bisa membantu penjual atau perusahaan untuk mengevaluasi masalah ataupun kendala dari produk mereka melalui ulasan misalnya, seperti Tokopedia, Shopee, dll.

Karena sifat data *text mining* yang tidak terstruktur, tentunya menjadi tantangan karena pengolahan data teks menjadi lebih sulit dibandingkan data terstruktur seperti *data warehouse*. Data yang tidak terstruktur ini biasanya kurang jelas, tidak konsisten, dan kontradiktif. Penggunaan kata-kata yang tidak sesuai

KBBI juga akan menyulitkan dalam proses pengolahan, misalnya: bahasa gaul, singkatan, slang, dll.

### 2.3 Analisis Sentimen

Analisis sentimen ialah tahap dari *text mining* yang mendefinisikan pendapat, *feeling* dan sikap yang terdapat di dalam teks. Hal ini digunakan dalam *marketing* untuk menganalisis misalnya komentar para peselancar atau perbandingan dan pengunjian para blogger. Analisis sentimen dapat dikategorikan ke dalam identifikasi emosi pada tingkat abstrak yang lebih rendah. Umumnya, sentimen bisa positif atau negative [10].

### 2.4 Cyberbullying

Pengertian *cyberbullying* menurut Patchin dan Hinduja (2015), *cyberbullying* merupakan perilaku yang disebabkan media teks elektronik atau internet yang dilakukan secara sengaja dan berulang. Tak hanya itu, menurut Rastati (2016) bahwa menyebarkan rumor tentang seseorang, mengintai, atau mengancam seseorang menggunakan berbagai jenis media elektronik bisa dikategorikan kedalam *cyberbullying*.

Menurut Manuel F. Lopez-Vizcaino, dkk. di jurnalnya berjudul “Early detection of cyberbullying on social media networks” menyebutkan bahwa *cyberbullying* merupakan masalah serius untuk masyarakat yang memiliki efek negatif sangat besar kepada korban, dapat merusak karena tingginya frekuensi dan penyebaran oleh teknologi informasi. Maka dari itu diperlukan deteksi dini di media sosial untuk memitigasi dampaknya terhadap para korban.

*Cyberbullying* bisa berdampak pada korban dengan aspek yang berbeda-beda, tidak hanya kesehatan, ada banyak lagi yang akan membawa kehidupan korban bahkan hingga ancaman. *Cyberbullying* tidak dapat dihindari oleh seluruh manusia di dunia, akan tetapi dapat dicegah.

### 2.5 Media Sosial

Media sosial merupakan media *online* yang dipakai untuk keperluan komunikasi, bertukar informasi jarak jauh sela satu *user* dengan *user* lainnya, dan

bisa diakses kapanpun dan dimanapun. Agar media sosial yang digunakan lancar, maka diperlukan koneksi internet yang stabil dan cepat. Internet muncul pada awal tahun 1970 dengan munculnya *Automatic Radar Plotting Aids* (ARPA) milik sistem Departemen Pertahanan Amerika Serikat yang difokuskan untuk menjaga jaringan komputer terhadap serangan nuklir karena Pentagon tidak mau kehilangan data dan sistem komunikasi yang dibangun menjadi hancur.

Kemunculan media sosial dimulai sejak akhir abad ke-19, tepatnya pada tahun 1970-an, dimana awalnya ditemukan papan bulletin yang mengizinkan untuk bisa berkomunikasi dengan orang lain melalui surat elektronik, mengupload dan mendownload software, semua ini dilakukan memakai modem yang telah disambungkan saluran telepon. Pada tahun 1995 lahir situs GeoCities yang menjalani web hosting, dan ini juga merupakan awal berdirinya website. Tahun 1997-1999 lahir media sosial pertama kali yaitu *Sixdegree.com* dan *Classmates.com*, dan juga muncul website untuk menciptakan *personal blog*, yaitu Blogger. Di tahun 2002, Friendster menjadi medsos yang sangat *booming*. Pada tahun 2003 hingga sekarang sudah banyak lahir berbagai jenis media sosial dengan ciri khas dan juga keunggulannya masing-masing.

## **2.6 Instagram**

Instagram adalah salah satu aplikasi media sosial yang tercipta dari salah satu perusahaan bernama Burbn, Inc. yang didirikan pada 6 Oktober 2010 oleh Kevin Systrom dan Mike Krieger yang sekarang menjabat CEO Instagram. Awalnya Kevin Systrom memiliki bakat di dunia teknologi karena sang ibu, Diane yang sudah lama bekerja di salah satu perusahaan teknologi dan periklanan yaitu monster.com. Aplikasi Instagram pertama kali dirilis untuk iOS user di App Store dan mendapatkan 100.000+ *user* di minggu pertama perilisan atau 1.000.000 *user* hingga Desember 2010.

Popularitas Instagram semakin melonjak dan membuat banyak investor menanamkan dana besar untuk mengembangkan aplikasi Instagram. Akhirnya pada bulan April 2012, berselang 2 minggu setelah perilisan versi Android secara resmi, popularitas Instagram semakin melonjak. Seminggu setelahnya, Instagram

diakuisisi oleh Facebook sebesar USD 1 Miliar. Lalu pada tahun 2018, Kevin Systrom dan Mike Krieger secara resmi mengundurkan diri.

Sekarang pengguna Instagram semakin banyak, apalagi saat ini fitur di Instagram semakin banyak, seperti: post video atau foto, user bisa memberikan like, komen, share dan save post tersebut. DM yang digunakan untuk chat secara pribadi. Sekarang Instagram juga menyediakan fitur Instastory dan Reels (dimana reels ini mirip seperti TikTok).

## 2.7 Teks Pre-processing

Teks pre-processing merupakan tahapan untuk mereduksi beberapa susunan bentuk kalimat menjadi satu kata. Tujuan utamanya adalah untuk memperoleh fitur-fitur inti dari kumpulan dokumen data yang telah dikumpulkan untuk meningkatkan relevansi antara kata dan *class* serta relevansi antara kata dan dokumen [11].

Tahapan *text pre-processing* ini ada 6 seperti **Gambar 2.1** Seperti berikut:



**Gambar 2.1** Tahapan *pre-processing*

### 2.7.1 Case Folding

Tahapan *case folding* digunakan untuk merubah semua kata dalam dokumen menjadi *lower case* semua.

### 2.7.2 Cleaning

Tahapan *cleaning* membersihkan data dari faktor-faktor yang tidak ada kaitannya dengan keterangan yang terdapat dalam dokumen, seperti karakter atau

symbol (“\$%^&():<>?!~[]”), angka, *link ur l*(<http://tokopedia.com>), *hashtag* (#), dan *mention* (@raisa6690).

### 2.7.3 Normalisasi

Tahapan normalisasi merupakan proses pengembalian dari kata tidak baku ke kata baku sesuai ketentuan KBBI (Kamus Besar Bahasa Indonesia). Misalnya penggunaan kata slang, singkatan, dan bahasa gaul.

### 2.7.4 Stopword Removal

Tahapan *stopword removal* membuang kata yang tidak berguna dan tidak memiliki makna di dalam dokumen, misalnya: di, ke, dan, pada, yang, kepada, dll.

### 2.7.5 Stemming

Tahapan *stemming* merupakan proses pengubahan kata di dalam dokumen yang awalnya berimbuhan menjadi kata dasar.

### 2.7.6 Tokenizing

Tahapan *tokenizing* ini memisahkan kata dari kalimat yang terdapat pada dokumen yang mulanya berbentuk kalimat menjadi per kata dan menghilangkan tanda baca.

## 2.8 TF-IDF

TF-IDF adalah ukuran statistic dan skema *term-weighted* yang menyediakan model *bag-of-words* dengan informasi penting. Aspek yang berbeda diekstrak dari kumpulan data yang di proses, seperti kata kerja, kata sifat, dan kata benda. TF-IDF digunakan untuk mengevaluasi signifikan sebuah kata ke dokumen pada dataset. Setiap kata diberi bobot dalam dokumen. Pada tahap indexing, pembobotan dilakukan dengan mengubah kata menjadi vector.

## 2.9 Support Vector Machine

Metode *Support Vector Machine* merupakan metode yang digunakan untuk melakukan analisis sentiment pada penelitian ini. Hasil yang ditentukan dengan

metode ini adalah klasifikasi *class bullying* atau *non-bullying* dan juga untuk meningkatkan nilai performa akurasi.

*Support Vector Machine* termasuk kategori *computational learning* dalam *artificial intelligent*, yaitu teknologi *machine learning* yang dikembangkan pada pertengahan 1990-an. Jika dibandingkan dengan teknologi *tradisional learning*, SVM memiliki dasar teori yang kuat. SVM membuat dan memelihara catatan terbaik pada banyak masalah yang spesifik seperti *handwritten numeric recognition*, *text classification*, dll [12]. *Support Vector Machine* merupakan pengklasifikasian *machine learning* yang diawasi dan biasanya digunakan dalam klasifikasi data berupa teks. SVM dibangun dengan menghasilkan *hyperplane* pemisah pada atribut fitur 2 kelas, dimana jarak antara *hyperplane* dan titik data yang berdekatan dari setiap kelas dimaksimalkan [6]. Pada dasarnya, *Support Vector Machine* (SVM) menggunakan data latih untuk mempelajari fungsi klasifikasi dengan membagi celah yang paling baik memisahkan kasus positif dari kasus negatif [13]. SVM memaksimalkan margin, yang merupakan jarak pemisah antara kelas data. SVM juga mampu bekerja pada dataset dengan dimensi tinggi.

Prinsip dasar SVM yaitu klasifikasi *linear* dan dikembangkan supaya bisa mengerjakan masalah *non-linear* dengan memasukkan konsep kernel trick pada ruang kerja dengan dimensi tinggi. Ada beberapa kernel yang sering digunakan, antara lain sebagai berikut [14]:

a. Linear

Persamaan dapat dirumuskan (1):

$$K(X_1, X_2) = X_1 \cdot X_2$$

b. Radial Basis Function

Persamaan RBF dapat dituliskan sebagai berikut (2):

$$K(X_1, X_2) = \exp(-\gamma \|X_1 - X_2\|^2)$$

c. Sigmoid

Rumus sigmoid (3):

$$K(X_1, X_2) = \tanh(\gamma X_1 \cdot X_2 + c)$$

## 2.10 Pengukuran Performa

Pada penelitian ini akan dilakukan pengujian dengan perhitungan nilai *precision*, *recall*, dan *accuracy* yang ditunjukkan pada persamaan tabel berikut:

**Tabel 2.2** *Confusion matrix*

		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	TP (True Positive)	FP (False Positive)
	FALSE	FN (False Negative)	TN (True Negative)

Uji deteksi adalah tahap yang digunakan untuk mengukur nilai performa dari sistem. Pengujian dilakukan dengan melakukan perbandingan performa matriks. Dengan melakukan pengukuran performa menggunakan rumus *precision*, *recall*, dan *accuracy*.

### 2.10.1 Precision

Presisi adalah metode pengukuran performa dengan menghitung perbandingan antara nilai *True Positive* (TP) dengan banyak data yang diprediksi bernilai positif. Rumus *precision* dapat ditulis sebagai berikut:

$$Precision = \frac{TP}{TP + FP}$$

### 2.10.2 Recall

Recall merupakan metode pengukuran performa yang menilai prediksi benar positif tentang keseluruhan data yang bernilai benar positif. Rumus perhitungan *recall* ditulis sebagai berikut:

$$Recall = \frac{TP}{TP + FN}$$

### 2.10.3 Accuracy

Accuracy adalah rasio prediksi pada hasil pengukuran kelengkapan dari seluruh data. Rumus persamaan *accuracy* ditulis sebagai berikut:

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



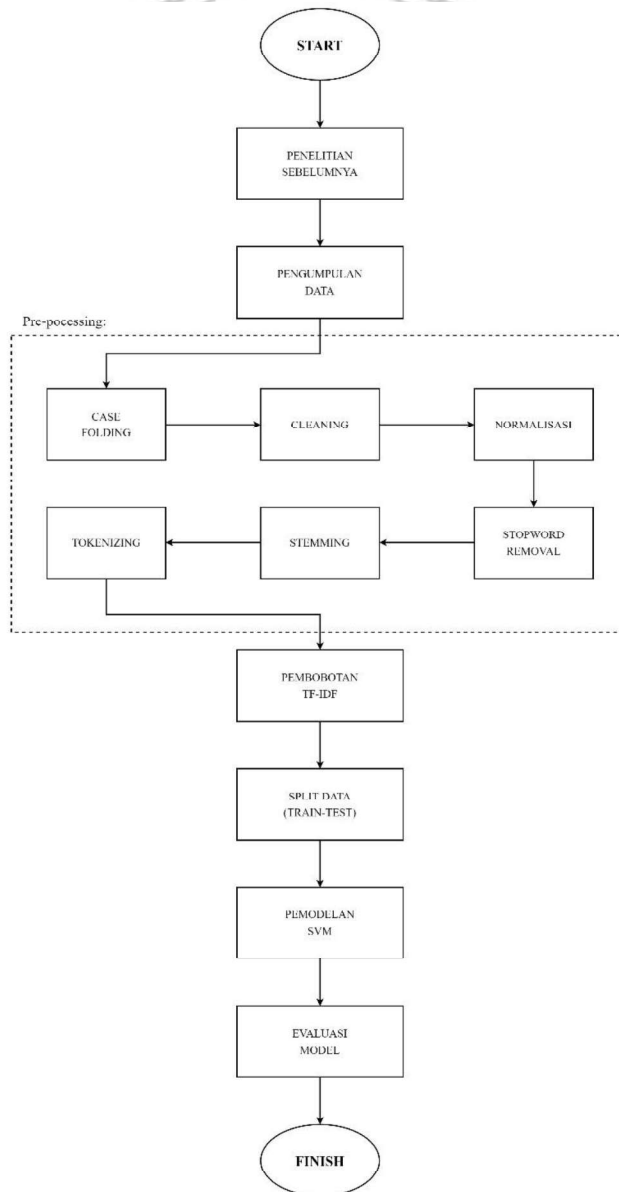
## BAB III

### METODOLOGI PENELITIAN

Pada bab ini berisikan alur metodologi untuk mencapai tujuan hasil penelitian menggunakan metode *Support Vector Machine* (SVM).

#### 3.1 Rancangan Penelitian

Terdapat 7 tahapan utama yang digunakan untuk menyelesaikan penelitian ini, tahapannya dapat dilihat pada **Gambar 3.1** dibawah:



**Gambar 3.1** Alur penelitian

### 3.2 Pengumpulan Data

Pada tahapan ini, pengumpulan data dilakukan secara manual dengan mengunjungi profil Instagram dari artis/selebgram setelah itu memilih beberapa post (foto atau video) di feed mereka, lalu mengambil beberapa komentar yang dituliskan oleh netizen pada post tersebut. Pencarian dan pengumpulan dataset ini dilakukan selama kurun waktu 2-3 bulan.

Total data pada penelitian kali ini ada sebanyak 650 data yang kemudian dibagi menjadi 2 kategori, yaitu *bullying* dan *non-bullying*. Dimana jumlah data untuk masing-masing kategori ada sebanyak 375 data. Contoh dataset dapat dilihat pada **Tabel 3.1**:

**Tabel 3.1** Dataset

No.	Nama Instagram	Komentar	Kategori	Tanggal Upload	Akun Artis/Selebgram
1	@khanayarudinita	Makin jelek aja anaknya, padahal ibu ayahnya cakep	<i>Bullying</i>	22 Juni 2019	@tasyakamila
2	@reniaulia225	Kok anaknya kayak udah tua gitu ya makanya kk tasya	<i>Bullying</i>	22 Juni 2019	@tasyakamila
3	@ghy.14	kamu emang cocoknya jadi pengusaha kuliner udah fix. Semoha lancet terussss	<i>Non – bullying</i>	27 Desember 2020	@rosameldianti_
4	@bundafahryfahry	Udah gk jaman ngebully orang hey netizen mending ngaca dari	<i>Non-bullying</i>	28 Desember 2020	@rosameldianti_

		kekurangan sendiri			
5	@rani__ara	Suka heran sama yg ngehina fisik,, ternyata bener ya kita semua pendosa dengan cara kita sendiri,,	<i>Non-bullying</i>	26 Desember 2020	@rosameldianti_
6	@yudharamm	Udah gila sekarang ini orang wkwk, udah jomblo jadi psikolog gal aku, ya jadilah seperti itu	<i>Bullying</i>	24 Desember 2020	@lutfiagizal
7	@lennyliando	Ibu kamu pelakor plus hamil di luar nikah ya? Pantesan anaknya kyk gini modelannya	<i>Bullying</i>	16 Juni 2020	@listychanpokemon
8	@tetikasm7615	serem banget muka lo mel kayak ayam potong merah banget	<i>Bullying</i>	27 Desember 2020	@rosameldianti_
9	@ahmadnfsv	Damage nya cewek pintar gada lawan	<i>Non-bullying</i>	22 Maret 2021	@isyanasarasvati

10	@kadospesialmu.id	Fokus ke nagita pas mau minum masih sempet baca doa. Keren bgt hasil dari didikan orang tua nya	Non-bullying	31 Mei 2021	@raffinagita17
11	@swity_nananana	Sebelum aku Unfoll, aku mau bilang NAJISSSS	Bullying	28 Desember 2020	@rosameldianti_
12	@story_quotes_	Anjirlah muka lo kek kuntilanak malah sok cantik	Bullying	1 Juni 2021	@dennisechariesta
13	@ini.ataaaa	Mirip ama actor korea yg main di film reply 1988 yg jadi kaka nya jung hwan wkwk	Non-bullying	15 Maret 2021	@syifahadjureal
14	@kylagluh	Ka cipa tu uda cantik, baik, manis bgt pula, manisnya tu ngelebihin gula	Non-bullying	10 Juni 2020	@syifahadjureal
15	@meta_fira	Maap kalau gua sering nyamain lu sama dugong soalnya emang mirip	Bullying	22 Desember 2020	@rosameldianti_

### 3.3 Pre-processing

Tahap ini merupakan tahap untuk mereduksi beberapa susunan bentuk kata menjadi satu kata. Tujuan utama dari pre-processing yaitu untuk mendapatkan kata kunci dari kumpulan dokumen data untuk meningkatkan relevansi antara kata dan dokumen serta relevansi antara kata dan kategori. Berikut tahapan pada *pre-processing*:

#### 3.3.1 Case Folding

Pada tahapan ini bertujuan untuk mengubah semua kata pada dokumen menjadi huruf kecil, contoh proses pada tahap ini dapat dilihat pada **Tabel 3.2** :

**Tabel 3.2** Proses *case folding*

Proses Case Folding	
Kalimat Awal	Setelah Diproses
AMAZING ISYANA!! Jujur aku amazed banget dengan skill dan minat Isyana dalam bermusik. Ga cuma bernyanyi tapi main alat musik pun jago banget	<b>amazing isyana!! jujur</b> aku amazed banget dengan skill dan minat <b>isyana</b> dalam bermusik. <b>ga</b> cuma bernyanyi tapi main alat musik pun jago banget
Aku doakan perempuan baik yg bernama Nagita selalu sehat, bahagia lahir batin, dan banyak yang sayang sm dia..	aku doakan perempuan baik yg bernama <b>nagita</b> selalu sehat, bahagia lahir batin, dan banyak yang sayang sm dia..
Congrats ka @isyanasarasvati, terus berkarya, makin sukses... dan terus mnjd inspirasi kita semua...	congrats ka @isyanasarasvati, terus berkarya, makin sukses... dan terus mnjd inspirasi kita semua...

#### 3.3.2 Cleaning

Pada tahapan ini bertujuan untuk membersihkan data dari komponen yang tidak memiliki informasi pada dokumen, seperti karakter, angka, *link url*, *hashtag*, dan *mentioned*. Contoh proses *cleaning* dari **Tabel 3.2** dapat dilihat pada **Tabel 3.3**:

**Tabel 3.3** Proses *cleaning*

Proses <i>Cleaning</i>	
Kalimat Awal	Setelah Diproses
amazing isyana!! jujur aku amazed banget dengan skill dan minat isyana dalam bermusik. ga cuma bernyanyi tapi main alat musik pun jago banget	amazing isyana jujur aku amazed banget dengan skill dan minat isyana dalam bermusik ga cuma bernyanyi tapi main alat musik pun jago banget
aku doakan perempuan baik yg bernama nagita selalu sehat, bahagia lahir batin, dan banyak yang sayang sm dia..	aku doakan perempuan baik yg bernama nagita selalu sehat bahagia lahir batin dan banyak yang sayang sm dia
congrats ka @isyanasarasvati, terus berkarya, makin sukses... dan terus mnjd inspirasi kita semua...	congrats ka isyanasarasvati terus berkarya makin sukses dan terus mnjd inspirasi kita semua

### 3.3.3 Normalisasi

Pada tahapan ini kata dalam dokumen yang tadinya tidak baku diubah menjadi kata baku sesuai aturan pada KBBI (Kamus Besar Bahasa Indonesia). Contoh proses normalisasi bisa dilihat pada **Tabel 3.4.**:

**Tabel 3.4** Proses normalisasi

Proses Normalisasi	
Kalimat Awal	Setelah Diproses
amazing isyana jujur aku amazed banget dengan skill dan minat isyana dalam bermusik ga cuma bernyanyi tapi main alat musik pun jago banget	amazing isyana jujur <b>saya</b> amazed <b>sekali</b> dengan skill dan minat isyana dalam bermusik <b>tidak</b> cuma bernyanyi <b>tetapi</b> main alat musik <b>juga</b> jago <b>sekali</b>
aku doakan perempuan baik yg bernama nagita selalu sehat	<b>saya</b> doakan perempuan baik <b>yang</b> bernama nagita selalu sehat

bahagia lahir batin dan banyak yang sayang sm dia	bahagia lahir batin dan banyak yang sayang <b>sama</b> dia
congrats ka isyanasarasvati terus berkarya makin sukses dan terus mnjd inspirasi kita semua.	congrats <b>kak</b> isyanasarasvati terus berkarya <b>semakin</b> sukses dan terus <b>menjadi</b> inspirasi kita semua

### 3.3.4 Stopword Removal

Tujuan dari tahapan ini adalah untuk menghapus kata di dalam dokumen yang tidak memiliki makna, misalnya: di, ke, dari, dan, dll. Contoh proses *stopword removal* bisa kita lihat pada **Tabel 3.5** dibawah ini:

**Tabel 3.5** Proses *stopword removal*

Proses <i>Stopword Removal</i>	
Kalimat Awal	Setelah Diproses
amazing isyana jujur saya amazed sekali dengan skill dan minat isyana dalam bermusik tidak cuma bernyanyi tetapi main alat musik juga jago sekali	amazing isyana jujur amazed skill minat isyana bermusik bernyanyi main alat musik jago
saya doakan perempuan baik yang bernama nagita selalu sehat bahagia lahir batin dan banyak yang sayang sama dia	doakan perempuan bernama nagita sehat bahagia lahir batin sayang
congrats kak isyanasarasvati terus berkarya semakin sukses dan terus menjadi inspirasi kita semua	congrats kak isyanasarasvati berkarya sukses inspirasi

### 3.3.5 Stemming

Pada tahapan ini kata di dalam dokumen yang awalnya berimbuhan diubah menjadi kata dasar. Contoh proses *stemming* dapat dilihat pada **Tabel 3.6** dibawah ini:

**Tabel 3.6** Proses *stemming*

Proses <i>Stemming</i>	
Kalimat Awal	Setelah Diproses
amazing isyana jujur amazed skill minat isyana bermusik bernyanyi main alat musik jago	amazing isyana jujur amazed skill minat isyana <b>musik nyanyi</b> main alat musik jago
doakan perempuan bernama nagita sehat bahagia lahir batin sayang	<b>doa</b> perempuan <b>nama</b> nagita sehat bahagia lahir batin sayang
congrats kak isyanasarasvati berkarya sukses inspirasi	congrats kak isyanasarasvati <b>karya</b> sukses inspirasi

### 3.3.6 Tokenizing

Pada proses ini, dokumen yang awalnya berupa kalimat akan diubah menjadi kata. Contoh tahapan *tokenizing* dapat dilihat pada **Tabel 3.7** dibawah ini:

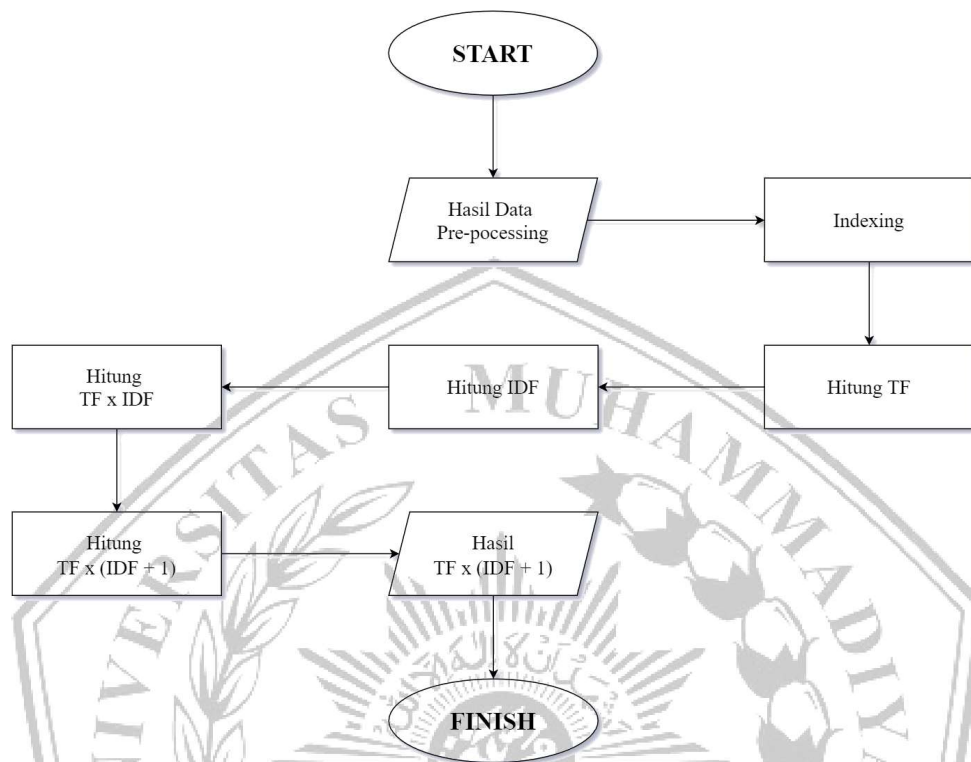
**Tabel 3.7** Proses *tokenizing*

Proses <i>Tokenizing</i>	
Kalimat Awal	Setelah Diproses
amazing isyana jujur amazed skill minat isyana musik nyanyi main alat musik jago	'amazing', 'isyana', 'jujur', 'amazed', 'skill', 'minat', 'isyana', 'musik', 'nyanyi', 'main', 'alat', 'musik', 'jago'
doa perempuan nama nagita sehat bahagia lahir batin sayang	'doa', 'perempuan', 'nama', 'nagita', 'sehat', 'bahagia', 'lahir', 'batin', 'sayang'
congrats kak isyanasarasvati karya sukses inspirasi	'congrats', 'kak', 'isyanasarasvati', 'karya', 'sukses', 'inspirasi'

### 3.4 Pembobotan TF-IDF

TF-IDF digunakan untuk mengevaluasi signifikansi sebuah kata ke dokumen dalam dataset [15], tiap kata dalam dokumen diberi bobot. Tahapan

perhitungan TF-IDF dapat dilihat pada **Gambar 3.2**, contoh dokumen terdapat pada **Tabel 3.8**



**Gambar 3.2** Perhitungan TF-IDF

**Tabel 3.8** Komentar Instagram

Komentar	
D1	amazing isyana jujur amazed skill minat isyana musik nyanyi main alat musik jago
D2	doa perempuan nama nagita sehat bahagia lahir batin sayang
D3	congrats kak isyanasarasvati karya sukses inspirasi

Sebelum perhitungan, kita melakukan konstruksi *inverted index* dengan *LEXICON*, prosesnya dapat kita lihat pada **Tabel 3.9** dibawah ini:

**Tabel 3.9** *Inverted index*

LEXICON		Posting List <doc_id, ft>
Term	N	
alat	1	<D1, 11>
amazed	1	<D1, 4>
amazing	1	<D1, 1>
bahagia	1	<D2, 6>
batin	1	<D2, 8>
congrats	1	<D3, 1>
doa	1	<D2, 1>
inspirasi	1	<D3, 6>
isyana	2	<D1, 2> <D1, 7>
isyanasarasvati	1	<D3, 3>
jago	1	<D1, 13>
jujur	1	<D1, 3>
kak	1	<D3, 2>
karya	1	<D3, 4>
lahir	1	<D2, 7>
main	1	<D1, 10>
minat	1	<D1, 6>
musik	2	<D1, 8> <D1, 12>
nagita	1	<D2, 4>
nama	1	<D2, 3>
nyanyi	1	<D1, 9>
perempuan	1	<D2, 2>
sayang	1	<D2, 9>
sehat	1	<D2, 5>
skill	1	<D1, 5>
sukses	1	<D3, 5>

Selanjutnya, kita lakukan perhitungan TF-IDF secara manual, proses perhitungan dapat dilihat pada **Tabel 3.10**:

**Tabel 3.10** Perhitungan TF-IDF

Token	tf			df	D/ df	idf = log (D/df)	tf x idf			tf x (idf + 1)		
	D1	D2	D3				D1	D2	D3	D1	D2	D3
alat	1	0	0	1	3	0.477	0.477	0	0	1.477	0	0
amazed	1	0	0	1	3	0.477	0.477	0	0	1.477	0	0
amazing	1	0	0	1	3	0.477	0.477	0	0	1.477	0	0
bahagia	0	1	0	1	3	0.477	0	0.477	0	0	1.477	0
batin	0	1	0	1	3	0.477	0	0.477	0	0	1.477	0
congrats	0	0	1	1	3	0.477	0	0	0.477	0	0	1.477
doa	0	1	0	1	3	0.477	0	0.477	0	0	1.477	0
inspirasi	0	0	1	1	3	0.477	0	0	0.477	0	0	1.477
isyana	2	0	0	2	1.5	0.176	0.352	0	0	2.352	0	0
isyana- sarasvati	0	0	1	1	3	0.477	0	0	0.477	0	0	1.477
jago	1	0	0	1	3	0.477	0.477	0	0	1.477	0	0
jujur	1	0	0	1	3	0.477	0.477	0	0	1.477	0	0
kak	0	0	1	1	3	0.477	0	0	0.477	0	0	1.477
karya	0	0	1	1	3	0.477	0	0	0.477	0	0	1.477
lahir	0	1	0	1	3	0.477	0	0.477	0	0	1.477	0
main	1	0	0	1	3	0.477	0.477	0	0	1.477	0	0
minat	1	0	0	1	3	0.477	0.477	0	0	1.477	0	0
musik	2	0	0	2	1.5	0.176	0.352	0	0	2.352	0	0
nagita	0	1	0	1	3	0.477	0	0.477	0	0	1.477	0
nama	0	1	0	1	3	0.477	0	0.477	0	0	1.477	0
nyanyi	1	0	0	1	3	0.477	0.477	0	0	1.477	0	0
perempuan	0	1	0	1	3	0.477	0	0.477	0	0	1.477	0
sayang	0	1	0	1	3	0.477	0	0.477	0	0	1.477	0
sehat	0	1	0	1	3	0.477	0	0.477	0	0	1.477	0
skill	1	0	0	1	3	0.477	0.477	0	0	1.477	0	0
sukses	0	0	1	1	3	0.477	0	0	0.477	0	0	1.477

### 3.5 Split Data

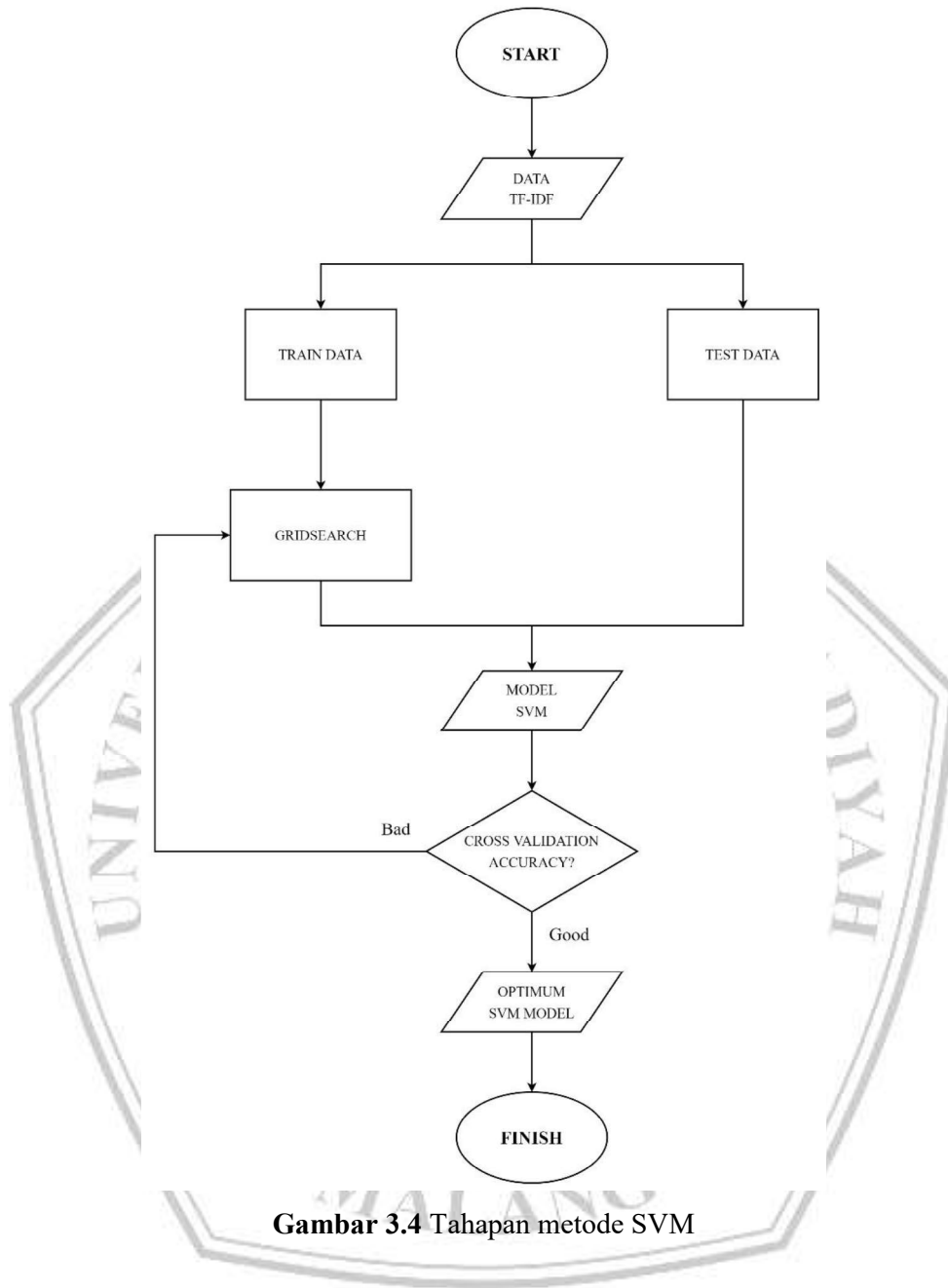
*Split data* bertujuan untuk memisahkan data pada metode yang digunakan menjadi 2 data, yaitu: *train data* dan *test data*. *Train data* dimasukkan ke dalam model dan *test data* dilakukan pengujian ke dalam model untuk menentukan nilai akurasi dari pengujian. Pada tahap ini, *test data* yang digunakan sebesar 10%, 20%, 30%, dan 40% dan pengambilan data dilakukan secara acak. Pada **Gambar 3.3** adalah *source code split data* menggunakan metode SVM.

```
▼ SPLIT DATA for TEST and TRAIN  
  
[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, stratify=y, random_state=42)  
    print(np.shape(X_train), np.shape(y_train), np.shape(X_test), np.shape(y_test))
```

**Gambar 3.3** *Split data Support Vector Machine*

### 3.6 Pemodelan SVM

Pada tahapan ini, metode yang digunakan adalah *Support Vector Machine*. Alur klasifikasi untuk komentar *cyberbullying* menggunakan metode SVM yaitu data dari TF-IDF dibagi menjadi 2 yaitu *train data* dan *test data*, data pada *train data* akan dilakukan Gridsearch untuk mengetahui parameter terbaik, setelah itu dilakukan pemodelan SVM, lalu perhitungan nilai akurasi cv (*cross validation*). Jika nilai akurasi jelek berarti hasil SVM kurang optimal, maka diulang lagi ke tahap gridsearch, jika hasil akurasi bagus maka hasil SVM sudah optimal. Tahapan pemodelan SVM dapat dilihat pada **Gambar 3.4** di bawah ini:



### 3.7 Evaluasi Model

Pada tahapan ini akan evaluasi model dengan menggunakan table *confusion matrix* untuk mengetahui nilai TP (*True Positive*), FP (*False Positive*), TN (*True Negative*), dan FN (*False Negative*) seperti yang terdapat pada **Tabel 3.11**, setelah itu dilakukan perhitungan nilai *precision*, *recall*, dan *accuracy*. Persamaan yang digunakan seperti yang tercantum pada tinjauan pustaka.

**Tabel 3.11** *Confusion matrix*

		Nilai Prediksi	
		TRUE	FALSE
Nilai Sebenarnya	TRUE	Komentar <i>non-bullying</i> Terprediksi <i>non-bullying</i> (True Positive)	Komentar <i>non-bullying</i> Terprediksi <i>bullying</i> (False Positive)
	FALSE	Komentar <i>bullying</i> Terprediksi <i>non-bullying</i> (False Negative)	Komentar <i>bullying</i> Terprediksi <i>bullying</i> (True Negative)



## BAB IV

### HASIL DAN PEMBAHASAN

Pada bab ini berisikan hasil dari penelitian pada komentar di Instagram menggunakan metode SVM. Isi pada bab ini yaitu implementasi dari tahapan yang sudah dijelaskan pada bab metodologi penelitian berbentuk potongan *source code* beserta hasil *output*.

#### 4.1 Implementasi Sistem

Untuk kelancaran penelitian dibutuhkan *hardware* dan *software* untuk mendukung proses penelitian.

##### 4.1.1 Hardware

Hardware merupakan perangkat/alat yang menunjang proses berjalannya *software* yang dipakai untuk penelitian ini, *hardware* yang digunakan seperti yang terlihat pada **Tabel 4.1** berikut:

**Tabel 4.1** *Hardware*

Hardware	Informasi Sistem
Processor	Intel® Core™ i3-5005U CPU @ 2.00GHz (\$ CPUs), ~2.0GHz
Motherboard	ASUSTeK COMPUTER INC. X455LAB
Memory	4GB RAM
Other	Keyboard

##### 4.1.2 Software

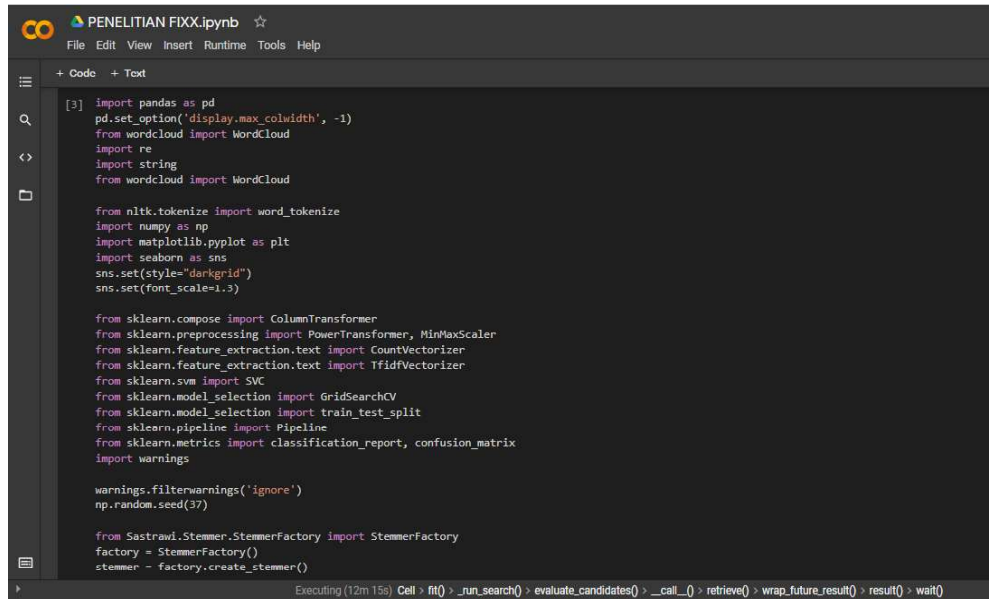
Software yang digunakan dalam mendukung penelitian tercantum pada **Tabel 4.2** dibawah:

**Tabel 4.2** *Software*

Software	Informasi
OS	Windows 10 Pro 64-bit (10.0, Build 19042)
Instagram	ver. 42.0.15.0
Ms. Excel	ver. 2016

## 4.2 Library

*Library* merupakan sekumpulan kode yang memiliki fungsi-fungsi tertentu dan dapat di *import/reuse* ke program lain yang digunakan untuk membangun dan mengembangkan *software*. *Library* yang dipakai pada penelitian ini bisa dilihat pada **Gambar 4.1** dibawah ini:

A screenshot of a Jupyter Notebook interface. The title bar shows 'PENELITIAN FIXX.ipynb'. The code cell contains the following Python code:

```
[3]: import pandas as pd
pd.set_option('display.max_colwidth', -1)
from wordcloud import WordCloud
import re
import string
from wordcloud import WordCloud

from nltk.tokenize import word_tokenize
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style='darkgrid')
sns.set(font_scale=1.3)

from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import PowerTransformer, MinMaxScaler
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report, confusion_matrix
import warnings

warnings.filterwarnings('ignore')
np.random.seed(37)

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()
```

The status bar at the bottom indicates 'Executing (12m 15s)' and shows a sequence of method calls: 'Cell', 'fit()', '\_run\_search()', 'evaluate\_candidates()', '.\_\_call\_\_()', 'retrieve()', 'wrap\_future\_result()', 'result()', and 'wait()'.

**Gambar 4.1** *Source code import library*

## 4.3 Dataset

Data pada penelitian ini berisikan data komentar *bullying* dan *non-bullying* pada Instagram dari artis/selebgram yang diambil secara manual dengan jumlah data untuk setiap kategorinya sebanyak 325 data dan disimpan dalam Ms. Excel dengan format *xlsx*. Data yang tadi disimpan kemudian di upload pada Google Drive supaya bisa dijalankan oleh Google Colab menggunakan *source code* yang ada pada **Gambar 4.2**.

```

[ ] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[ ] import os
os.environ = "/content/drive/My Drive/043_Sentiment_Analysis"

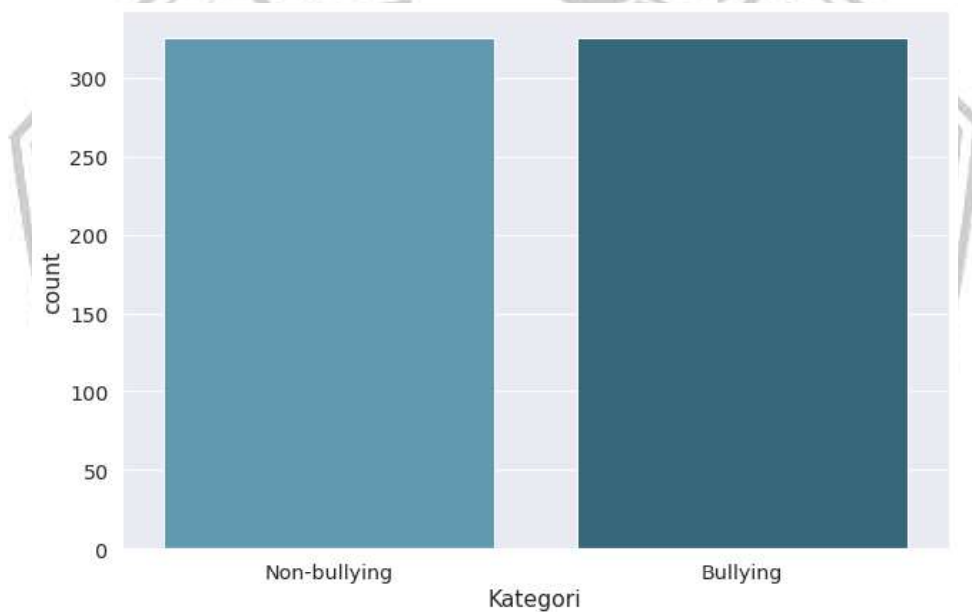
[ ] %cd /content/drive/My Drive/043_Sentiment_Analysis

/content/drive/My Drive/043_Sentiment_Analysis

[ ] data = pd.read_excel('DATASET CYBERBULLYING INSTAGRAM - FINAL.xlsx')
data = data[['Komentar', 'Kategori']]

```

**Gambar 4.2** *Syntax import* Google Drive pada Google Colab



**Gambar 4.3** *Dataset komentar cyberbullying* di Instagram

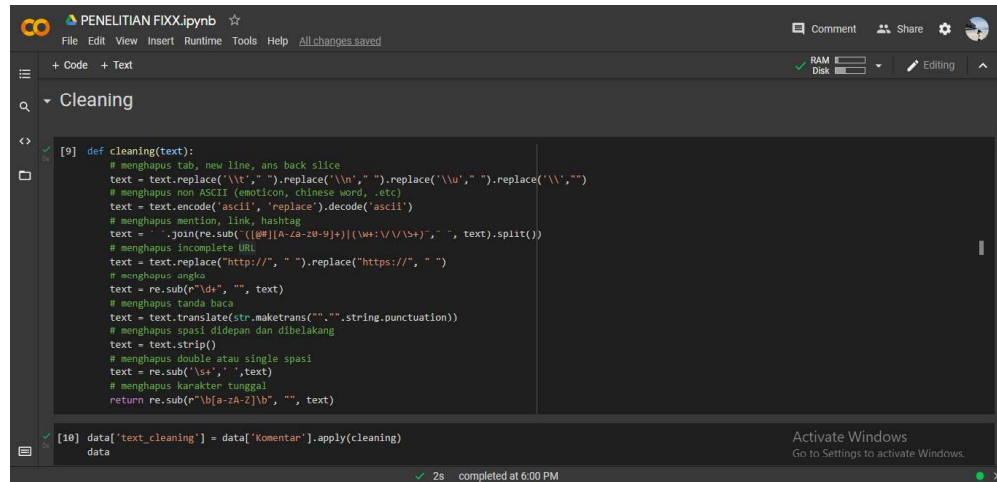
Pada **Gambar 4.3** menunjukan jumlah dataset untuk masing-masing *class* (*bullying* dan *non-bullying*), dimana jumlah datanya sama-sama 325 data.

#### 4.4 Pre-Processing

Pada tahapan ini akan dilakukan implementasi yang berguna membersihkan data dari kata-kata yang tidak memiliki makna. Tahapan praproses ada 6, yaitu: *cleaning*, *case folding*, *normalizing*, *stopword removal*, *stemming*, *tokenizing*.

#### 4.4.1 Cleaning

Menggunakan fungsi *def* diberi nama *cleaning* dimana mencakup 3 proses dengan memanfaatkan library *re*, *re.sub* syntax tercantum pada **Gambar 4.4**.



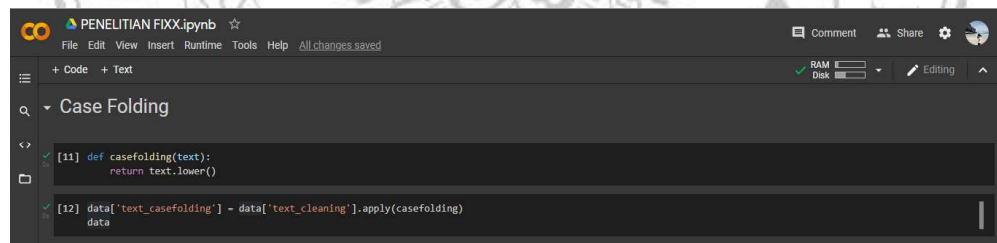
```
[9] def cleaning(text):
    # menghapus tab, new line, and back slice
    text = text.replace('\t',' ').replace('\n',' ').replace('\u',' ').replace('\\','')
    # menghapus non ASCII (emoticon, chinese word, etc)
    text = text.encode('ascii','replace').decode('ascii')
    # menghapus mention, link, hashtag
    text = ' '.join(re.sub('([#][A-Za-z0-9-]+)|(@+?[\w]+\s*)|(\w+:\w+/\w+)', '', text).split())
    # menghapus incomplete URL
    text = text.replace('http://', ' ').replace('https://', ' ')
    # menghapus angka
    text = re.sub(r'\d+', '', text)
    # menghapus tanda baca
    text = text.translate(str.maketrans('', '', string.punctuation))
    # menghapus spasi di depan dan dibelakang
    text = text.strip()
    # menghapus double atau single spasi
    text = re.sub('\s+', ' ', text)
    # menghapus karakter tunggal
    return re.sub(r'[a-zA-Z]\b', '', text)

[10] data['text_cleaning'] = data['komentar'].apply(cleaning)
data
```

**Gambar 4.4** Syntax tahapan *cleaning*

#### 4.4.2 Case Folding

Menggunakan fungsi *def* diberi nama *casefolding* untuk mengubah semua huruf capital menjadi *lowercase*, *syntax* dapat dilihat pada **Gambar 4.5**.



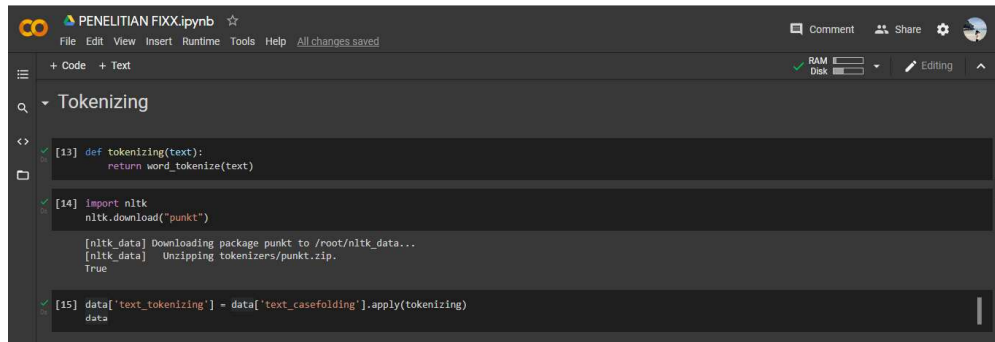
```
[11] def casefolding(text):
    return text.lower()

[12] data['text_casefolding'] = data['text_cleaning'].apply(casefolding)
data
```

**Gambar 4.5** Syntax tahapan *case folding*

#### 4.4.3 Tokenizing

Menggunakan fungsi *def* diberi nama *tokenizing* untuk memisahkan kalimat menjadi kata, *library* yang digunakan adalah *nltk*, *syntax* seperti di **Gambar 4.6**.

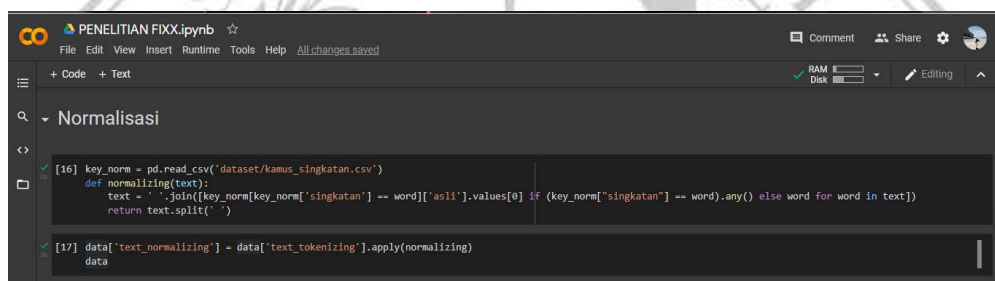
The screenshot shows a Jupyter Notebook interface with the title 'PENELITIAN FIXX.ipynb'. The code is organized into three cells under the heading 'Tokenizing'. The first cell defines a function 'tokenizing(text)' that returns 'word\_tokenize(text)'. The second cell imports 'nltk' and downloads the 'punkt' tokenizer. The third cell applies the 'tokenizing' function to a dataset column 'text\_casefolding' and assigns the result to 'data['text\_tokenizing']'.

```
[13] def tokenizing(text):  
      return word_tokenize(text)  
  
[14] import nltk  
      nltk.download("punkt")  
  
[nltk_data] Downloading package punkt to /root/nltk_data...  
[nltk_data] Unzipping tokenizers/punkt.zip.  
True  
  
[15] data['text_tokenizing'] = data['text_casefolding'].apply(tokenizing)  
      data
```

Gambar 4.6 Syntax tahapan *tokenizing*

#### 4.4.4 Normalisasi

Menggunakan fungsi *def* diberi nama *normalizing* untuk mengubah kata-kata singkatan menjadi kata sesuai dengan ketentuan KBBI yang tersimpan di dalam *kamus\_singkatan.csv*. Syntax seperti pada Gambar 4.7.

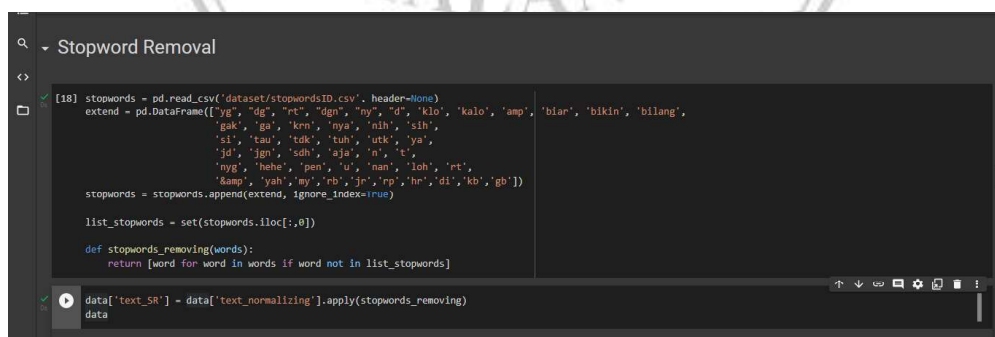
The screenshot shows a Jupyter Notebook interface with the title 'PENELITIAN FIXX.ipynb'. The code is organized into two cells under the heading 'Normalisasi'. The first cell reads a CSV file 'kamus\_singkatan.csv' into a DataFrame 'key\_norm' and defines a function 'normalizing(text)' that replaces abbreviations with their full forms from the DataFrame. The second cell applies the 'normalizing' function to the 'text\_tokenizing' column of the dataset and assigns the result to 'data['text\_normalizing']'.

```
[16] key_norm = pd.read_csv('dataset/kamus_singkatan.csv')  
      def normalizing(text):  
          text = ' '.join([key_norm[key_norm['singkatan'] == word][0].values[0] if (key_norm['singkatan'] == word).any() else word for word in text])  
          return text.split(" ")  
  
[17] data['text_normalizing'] = data['text_tokenizing'].apply(normalizing)  
      data
```

Gambar 4.7 Syntax tahapan normalisasi

#### 4.4.5 Stopword Removal

Menggunakan fungsi *def* dan diberi nama *stopwords\_removal* untuk menghapus kata yang tidak memiliki makna yang tersimpan didalam *stopwordsID.csv*. Source code seperti Gambar 4.8.

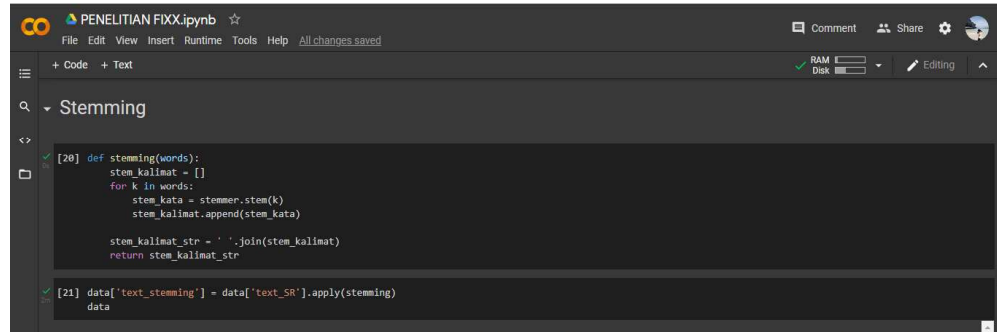
The screenshot shows a Jupyter Notebook interface with the title 'PENELITIAN FIXX.ipynb'. The code is organized into two cells under the heading 'Stopword Removal'. The first cell reads a CSV file 'stopwordsID.csv' into a DataFrame 'stopwords' and defines a function 'stopwords\_removal(words)' that removes words from the input list if they are in the 'stopwords' DataFrame. The second cell applies the 'stopwords\_removal' function to the 'text\_normalizing' column of the dataset and assigns the result to 'data['text\_SR']'.

```
[18] stopwords = pd.read_csv('dataset/stopwordsID.csv', header=None)  
      extend = pd.DataFrame({'yg', 'dg', 'rt', 'dgn', 'ny', 'd', 'kalo', 'amp', 'bian', 'bikin', 'bilang',  
                             'gak', 'ga', 'krn', 'nya', 'nih', 'sih',  
                             'si', 'tau', 'tdk', 'tuh', 'utk', 'ya',  
                             'jd', 'jgn', 'sdh', 'aja', 'a', 't',  
                             'nyg', 'hehe', 'pen', 'u', 'nan', 'loh', 'rt',  
                             '&amp;', 'yah', 'my', 'rb', 'jn', 'rp', 'hr', 'di', 'kb', 'gb'})  
      stopwords = stopwords.append(extend, ignore_index=True)  
  
      list_stopwords = set(stopwords.iloc[:,0])  
  
      def stopwords_removal(words):  
          return [word for word in words if word not in list_stopwords]  
  
data['text_SR'] = data['text_normalizing'].apply(stopwords_removal)  
data
```

Gambar 4.8 Syntax tahapan *stopword removal*

#### 4.4.6 Steming

Menggunakan fungsi *def* dan diberi nama *steming* untuk merubah kata imbuhan ke kata dasar. *Source code* seperti yang tercantum pada **Gambar 4.9**.



```
[20] def stemming(words):
      stem_kalimat = []
      for k in words:
          stem_kata = stemmer.stem(k)
          stem_kalimat.append(stem_kata)

      stem_kalimat_str = ' '.join(stem_kalimat)
      return stem_kalimat_str

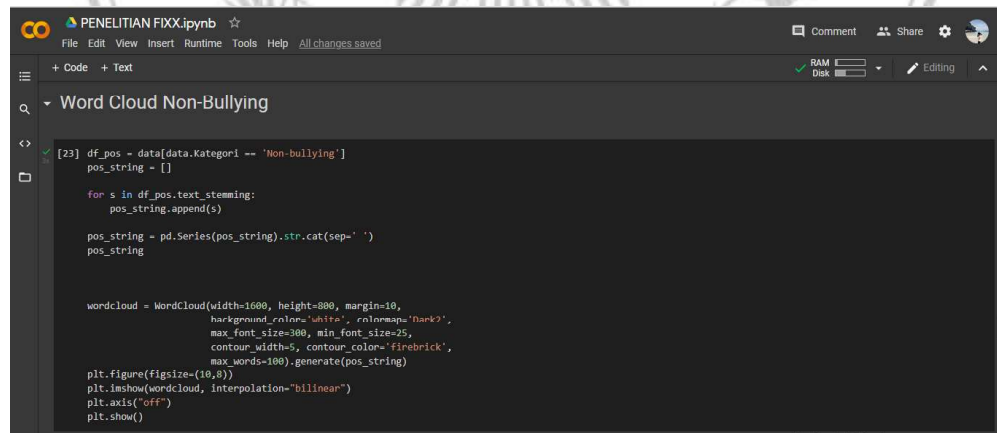
[21] data['text_stemming'] = data['text_SR'].apply(steming)
      data
```

**Gambar 4.9** *Syntax tahapan steming*

#### 4.5 Word Cloud

*Word cloud* berisi frekuensi kata-kata yang sering muncul pada suatu kumpulan teks dalam dokumen. Frekuensi kemunculan sebuah kata dapat dilihat dari ukurannya, semakin sering kata itu muncul maka semakin besar ukurannya dan juga sebaliknya.

*Source code word cloud* seperti yang terlihat pada **Gambar 4.10** dan **Gambar 4.11**, sedangkan *output word cloud* bisa dilihat pada **Gambar 4.12** dan **Gambar 4.13**.



```
[23] df_pos = data[data.Kategori == 'Non-bullying']
      pos_string = []

      for s in df_pos.text_stemming:
          pos_string.append(s)

      pos_string = pd.Series(pos_string).str.cat(sep=' ')
      pos_string

      wordcloud = WordCloud(width=1600, height=800, margin=10,
                             background_color='white', colormap='dark2',
                             max_font_size=300, min_font_size=25,
                             contour_width=5, contour_color='firebrick',
                             max_words=100).generate(pos_string)

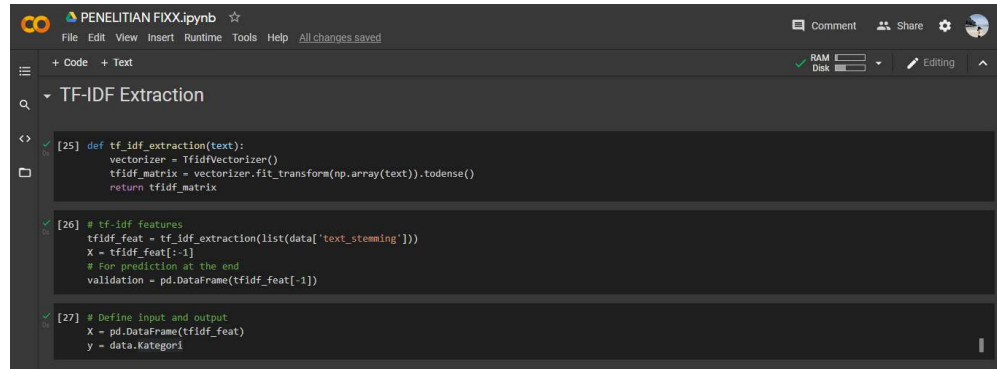
      plt.figure(figsize=(10,8))
      plt.imshow(wordcloud, interpolation='bilinear')
      plt.axis("off")
      plt.show()
```

**Gambar 4.10** *Source code word cloud non-bullying*



## 4.6 TF-IDF

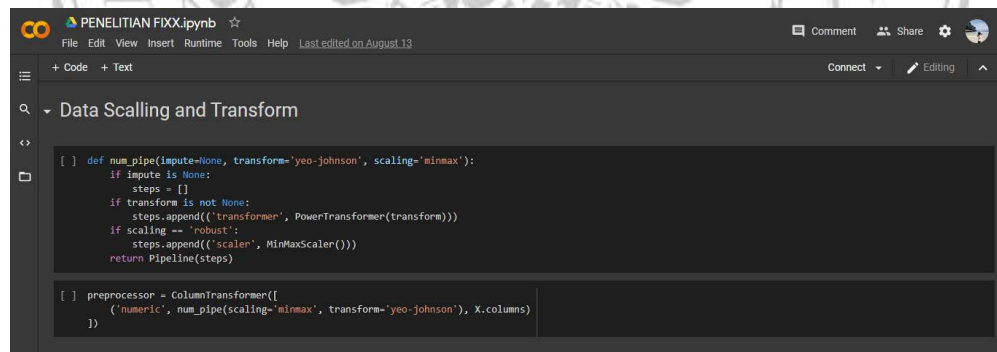
Mengsekstrak kata dari word cloud untuk dilakukan perhitungan pembobotan, data diambil dari hasil proses *steming*. *Syntax* seperti yang dapat dilihat pada **Gambar 4.14**.

The image shows a Jupyter Notebook interface with the title 'PENELITIAN FIXX.ipynb'. The code is organized into three cells. The first cell defines a function 'tf\_idf\_extraction(text)' that uses 'TfidfVectorizer' to create a 'tfidf\_matrix'. The second cell applies this function to a list of text from a dataset, creating 'tfidf\_feat', and then uses 'X = tfidf\_feat[-1]' and 'validation = pd.DataFrame(tfidf\_feat[-1])'. The third cell defines 'X = pd.DataFrame(tfidf\_feat)' and 'y = data.Kategori'.

**Gambar 4.14** *Source code TF-IDF*

## 4.7 Scalling and Transform

*Scalling* yang digunakan yaitu *MinMaxScaler* dan *transform* menggunakan metode *yeo-johnson*, lalu diterapkan pada dataset. Source code bisa dilihat pada **Gambar 4.15** dibawah:

The image shows a Jupyter Notebook interface with the title 'PENELITIAN FIXX.ipynb'. The code is in a single cell titled 'Data Scalling and Transform'. It defines a function 'num\_pipe' that takes 'impute', 'transform', and 'scaling' as arguments. Inside the function, it creates a list of steps: 'impute' (using 'SimpleImputer'), 'transform' (using 'PowerTransformer'), and 'scaling' (using 'MinMaxScaler'). It then returns a 'Pipeline' of these steps. Below the function, it creates a 'preprocessor' object using 'ColumnTransformer' to apply the 'num\_pipe' to the 'numeric' columns of the dataset 'X'.

**Gambar 4.15** *Syntax scalling and transform*

## 4.8 Pemodelan Support Vector Machine

### 4.8.1 Tunning Hyperparameter

Digunakan untuk menentukan parameter terbaik dari metode SVM, parameter pada penelitian ini menggunakan kernel, gamma, dan C. Pada kernel menggunakan rbf, linear, dan sigmoid, gamma menggunakan array dari 0.0001,

0.001, 0.01, dan 0.1, C menggunakan array dari 10, 100, 1000, hingga 10000. Source code seperti pada **Gambar 4.16** berikut:

```
[ ] svm_params = {'algo_kernel': ['rbf', 'linear', 'sigmoid'],
                  'algo_gamma': np.array([1.e-04, 1.e-03, 1.e-02, 1.e-01]),
                  'algo_C': np.array([1.e+01, 1.e+02, 1.e+03, 1.e+04])}
```

**Gambar 4.16** *Syntax tuning hyperparameter*

#### 4.8.2 Gridsearch CV

Dilakukan iterasi sebanyak 500 dan GridSearch pada *cross validation* sebanyak 5x. Hasil yang didapat yaitu skor *train data* 0.99 (99%) dan skor *test data* 0.86 (86%). *Source code* terlampir pada **Gambar 4.17**.

```
[ ] pipeline = Pipeline([
    ('prep', preprocessor),
    ('algo', SVC(max_iter=500))
])

model = GridSearchCV(pipeline, svm_params, cv=5, n_jobs=-1, verbose=1)
model.fit(X_train, y_train)

print(model.best_params_)
print(model.score(X_train, y_train), model.best_score_, model.score(X_test, y_test))
```

Fitting 5 folds for each of 48 candidates, totalling 240 fits  
[Parallel(n\_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.  
[Parallel(n\_jobs=-1)]: Done 46 tasks | elapsed: 3.4min  
[Parallel(n\_jobs=-1)]: Done 196 tasks | elapsed: 14.3min  
[Parallel(n\_jobs=-1)]: Done 240 out of 240 | elapsed: 17.4min finished  
{'algo\_C': 10.0, 'algo\_gamma': 0.0001, 'algo\_kernel': 'sigmoid'}  
0.9948717948717949 0.8290598290598291 0.8615384615384616

**Gambar 4.17** *Source code pemodelan SVM*

Hasil yang diperoleh dari masing-masing fold dan nilai accuracy tertinggi dapat dilihat di **Gambar 4.18** dibawah.

+ Code + Text				Connect + + +						
10	10.0	0.1000	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
43	10000.0	0.0100	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
7	10.0	0.0100	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
40	10000.0	0.1000	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
4	10.0	0.0010	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
34	1000.0	0.1000	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
36	10000.0	0.0001	rbf	0.769231	0.786325	0.846154	0.794872	0.786325	0.796581	
24	1000.0	0.0001	rbf	0.769231	0.786325	0.846154	0.794872	0.786325	0.796581	
20	100.0	0.0100	sigmoid	0.709402	0.735043	0.846154	0.743590	0.769231	0.737495	
8	10.0	0.0100	sigmoid	0.709402	0.735043	0.846154	0.743590	0.735043	0.735846	
35	1000.0	0.1000	sigmoid	0.726496	0.735043	0.846154	0.709402	0.735043	0.750427	
32	1000.0	0.0100	sigmoid	0.709402	0.726496	0.829060	0.726496	0.726496	0.743590	
47	10000.0	0.1000	sigmoid	0.726496	0.735043	0.846154	0.709402	0.692308	0.741880	
44	10000.0	0.0100	sigmoid	0.726496	0.735043	0.846154	0.709402	0.692308	0.741880	
23	100.0	0.1000	sigmoid	0.726496	0.683761	0.777778	0.709402	0.735043	0.726496	
51	10.0	0.1000	sigmoid	0.717949	0.692308	0.769231	0.735043	0.709402	0.724786	
30	1000.0	0.0100	rbf	0.598291	0.547009	0.598291	0.547009	0.564103	0.570940	
18	100.0	0.0100	rbf	0.598291	0.547009	0.598291	0.547009	0.564103	0.570940	
42	10000.0	0.0100	rbf	0.598291	0.547009	0.598291	0.547009	0.564103	0.570940	
6	10.0	0.0100	rbf	0.598291	0.547009	0.598291	0.547009	0.564103	0.570940	
33	1000.0	0.1000	rbf	0.504274	0.504274	0.504274	0.521368	0.512821	0.509402	
21	100.0	0.1000	rbf	0.504274	0.504274	0.504274	0.521368	0.512821	0.509402	
9	10.0	0.1000	rbf	0.504274	0.504274	0.504274	0.521368	0.512821	0.509402	
45	10000.0	0.1000	rbf	0.504274	0.504274	0.504274	0.521368	0.512821	0.509402	

Activate Windows  
Go to Settings to activate Windows.

+ Code + Text				Connect + + +						
algo_c	algo_gamma	algo_kernel	Accuracy CV1	Accuracy CV2	Accuracy CV3	Accuracy CV4	Accuracy CV5	Mean Accuracy		
2	10.0	0.0001	sigmoid	0.820513	0.777778	0.829060	0.871795	0.846154	0.829060	
0	10.0	0.0001	rbf	0.011966	0.777778	0.863248	0.829060	0.820513	0.820513	
14	100.0	0.0001	sigmoid	0.794872	0.777778	0.871795	0.794872	0.829060	0.813675	
3	10.0	0.0010	rbf	0.769231	0.777778	0.846154	0.837607	0.794872	0.805128	
26	1000.0	0.0001	sigmoid	0.769231	0.769231	0.863248	0.777778	0.837607	0.803419	
38	10000.0	0.0001	sigmoid	0.769231	0.769231	0.863248	0.777778	0.837607	0.803419	
12	100.0	0.0001	rbf	0.777778	0.786325	0.863248	0.794872	0.794872	0.803419	
41	10000.0	0.0010	sigmoid	0.769231	0.811966	0.863248	0.769231	0.803419	0.803419	
5	10.0	0.0010	sigmoid	0.760684	0.794872	0.800342	0.777778	0.803419	0.803419	
29	1000.0	0.0010	sigmoid	0.769231	0.811966	0.863248	0.769231	0.803419	0.803419	
17	100.0	0.0010	sigmoid	0.769231	0.811966	0.863248	0.769231	0.803419	0.803419	
27	1000.0	0.0010	rbf	0.777778	0.760684	0.846154	0.846154	0.777778	0.801709	
19	100.0	0.0100	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
25	10000.0	0.0001	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
1	10.0	0.0001	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
28	1000.0	0.0010	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
22	100.0	0.1000	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
37	10000.0	0.0001	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
15	100.0	0.0010	rbf	0.777778	0.760684	0.846154	0.846154	0.777778	0.801709	
39	10000.0	0.0010	rbf	0.777778	0.760684	0.846154	0.846154	0.777778	0.801709	
16	100.0	0.0010	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
31	1000.0	0.0100	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
13	100.0	0.0001	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	
40	10000.0	0.0010	linear	0.777778	0.760684	0.863248	0.786325	0.820513	0.801709	

Activate Windows  
Go to Settings to activate Windows.

**Gambar 4.18** Output CV result

Mekanisme pengujian *cross validation* yaitu *train data* dibagi menjadi k (nilai fold) subset (subhimpunan), masing-masing subset akan dijadikan *test data* dari klasifikasi dari k-1 subset lain. Nilai *error* dari masing-masing k test dihitung *mean*-nya.

## 4.8 Classification Report

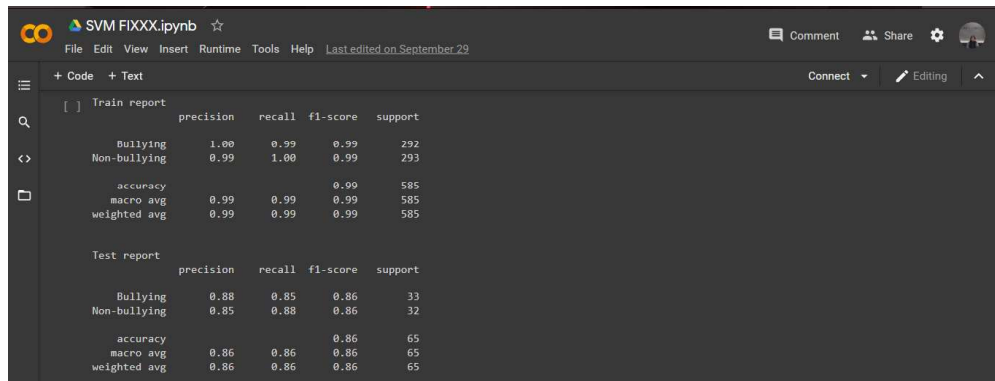
Berisi hasil *accuracy* dari *train data* dan *test data*. Untuk *syntax* dan *output* tercantum pada **Gambar 4.19** dan **Gambar 4.20** dibawah. Dapat dilihat bahwa *accuracy* untuk *train data* sebesar 1 (100%) dan *accuracy* untuk *test data* sebesar 0.85 (85%).

```

[ ] print("Train report")
print(classification_report(y_train, model.predict(X_train)))
print()
print("Test report")
print(classification_report(y_test, model.predict(X_test)))
print()

```

Gambar 4.19 Syntax classification report



Gambar 4.20 Output classification report

## 4.9 Confusion Matrix

Disini dilihat hasil yang sebelumnya kita prediksi apakah sesuai dengan kategorinya atau tidak, dan berapa banyak data yang kita prediksi untuk kategori *bullying* dan *non-bullying* justru terprediksi bukan dari kategori tsb. Syntax dan table *confusion matrix* dapat dilihat pada gambar dibawah:

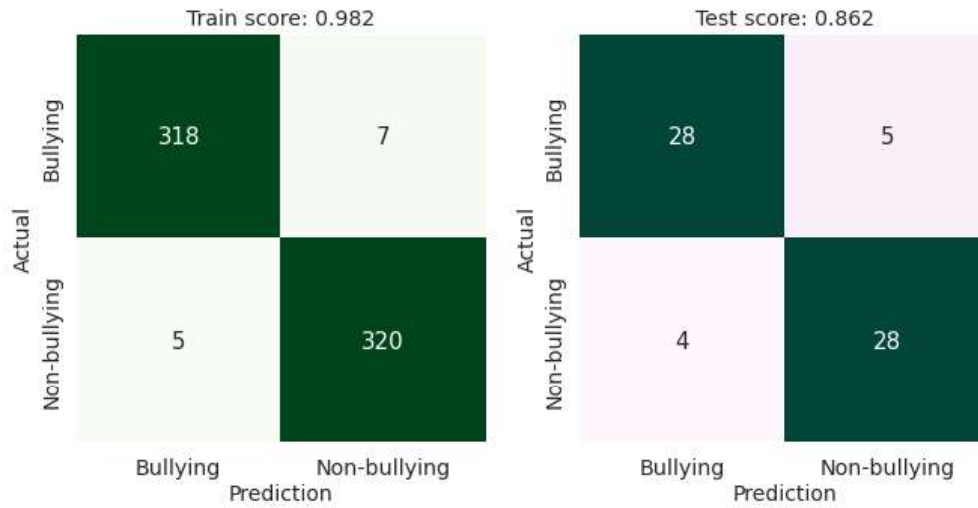
```

plt.figure(figsize=(11, 5))
plt.subplot(121)
labels = y_train.unique()
cm = confusion_matrix(y, model.predict(X), labels=labels)
sns.heatmap(cm, annot=True, square=True, cmap='Greens', cbar=False, xticklabels=labels, yticklabels=labels,
            fmt='d', annot_kws={"fontsize": 15})
plt.title(f'Train score: {model.score(X, y):.3f}', fontsize=14)
plt.xlabel('Prediction', fontsize=14)
plt.ylabel('Actual', fontsize=14)
plt.yticks(rotation=90, verticalalignment='center');

plt.subplot(122)
labels = y_test.unique()
cm = confusion_matrix(y_test, model.predict(X_test), labels=labels)
sns.heatmap(cm, annot=True, square=True, cmap='PuBuGn', cbar=False, xticklabels=labels, yticklabels=labels,
            fmt='d', annot_kws={"fontsize": 15})
plt.title(f'Test score: {model.score(X_test, y_test):.3f}', fontsize=14)
plt.xlabel('Prediction', fontsize=14)
plt.ylabel('Actual', fontsize=14)
plt.yticks(rotation=90, verticalalignment='center');

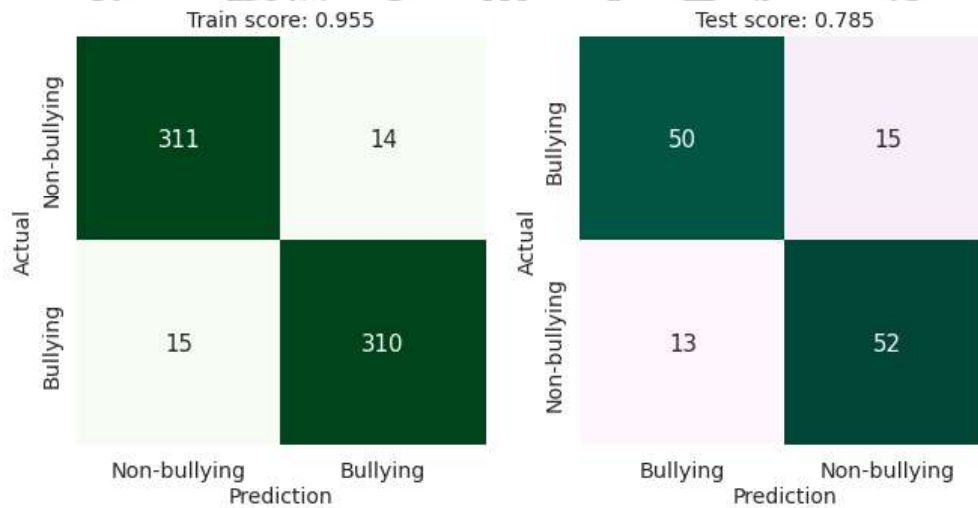
```

Gambar 4.21 Syntax confusion matrix



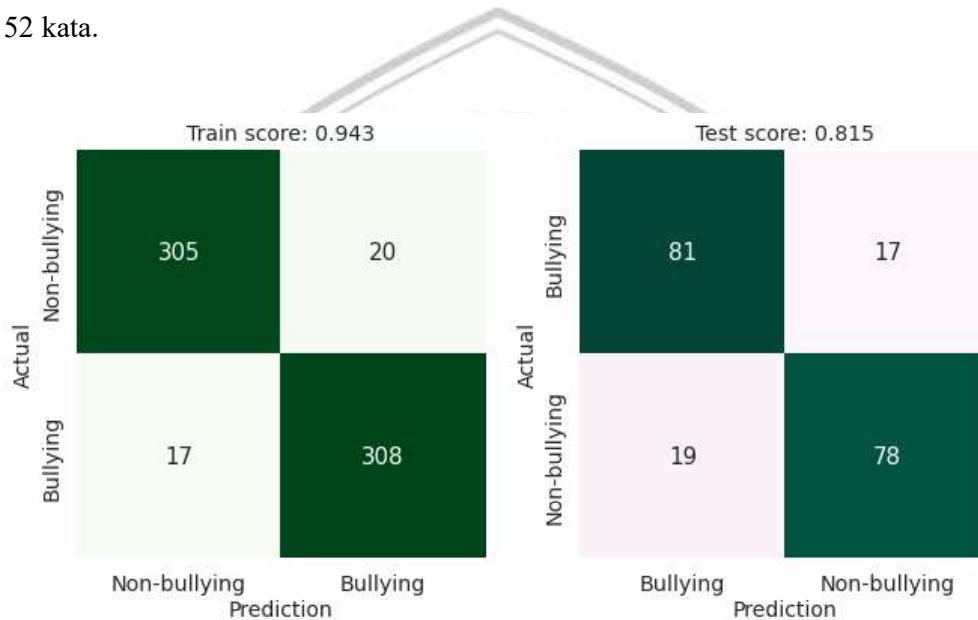
**Gambar 4.22** *Confusion matrix split data 90:10*

Dari *confusion matrix* pada **Gambar 4.22** dapat disimpulkan bahwa pada *train* yang sebenarnya *bullying* dan terprediksi *bullying* ada 318 data, yang sebenarnya *bullying* tetapi terprediksi *non-bullying* ada 7 data, yang sebenarnya *non-bullying* tetapi terprediksi *bullying* ada 5 data, yang sebenarnya *non-bullying* dan terprediksi *non-bullying* ada 320 data. Pada *test* yang sebenarnya *bullying* dan terprediksi *bullying* juga ada 28 data, yang sebenarnya *bullying* tetapi terprediksi *non-bullying* ada 5 data, yang sebenarnya *non-bullying* tetapi terprediksi *bullying* ada 4 data, yang sebenarnya *bullying* dan terprediksi *non-bullying* ada 28 data.



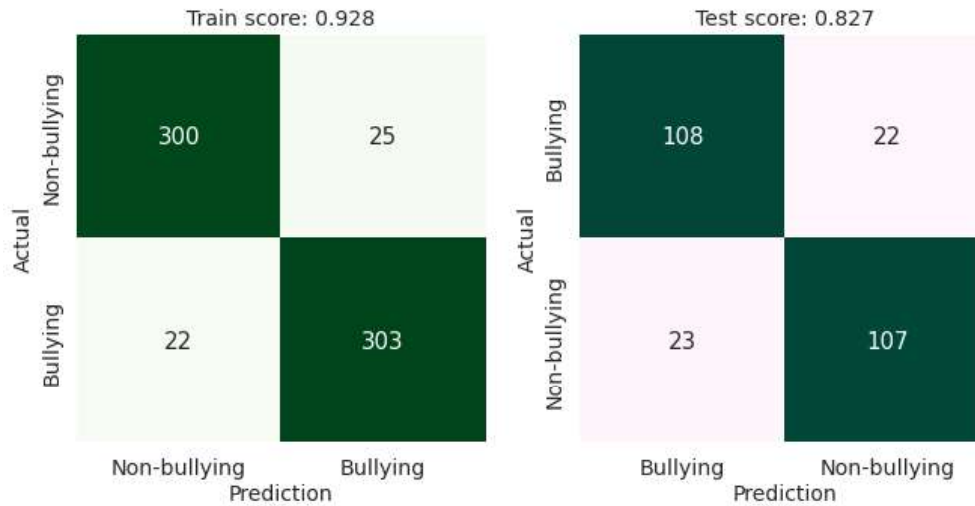
**Gambar 4.23** *Confusion matrix split data 80:20*

Dari **Gambar 4.23** didapatkan hasil pada *train* yang sebenarnya *non-bullying* dan terprediksi benar ada 311 data, yang sebenarnya *non-bullying* dan terprediksi salah ada 14 data, yang sebenarnya *bullying* dan terprediksi salah ada 15 data, yang sebenarnya *bullying* dan terprediksi benar ada 310 data. Pada *test* yang sebenarnya *bullying* dan terprediksi benar ada 50 data, yang sebenarnya *bullying* dan terprediksi salah ada 15 data, yang sebenarnya *non-bullying* dan terprediksi salah ada 13 data, yang sebenarnya *non-bullying* dan terprediksi benar ada 52 data.



**Gambar 4.24** *Confusion matrix split data 70:30*

Dari *confusion matrix* diatas dapat disimpulkan hasil dari *train* yang sebenarnya *non-bullying* dan terprediksi *non-bullying* ada 305 data, yang sebenarnya *non-bullying* tetapi terprediksi *bullying* ada 20 data, yang sebenarnya *bullying* tetapi terprediksi *non-bullying* ada 17 data, yang terprediksi *bullying* dan benar ada 308 data. Pada *test* yang sebenarnya *bullying* dan terprediksi *bullying* ada 81 data, yang sebenarnya *bullying* tetapi terprediksi *non-bullying* ada 17 data, yang sebenarnya *non-bullying* tetapi terprediksi *bullying* ada 19 data, yang sebenarnya *non-bullying* dan terprediksi *non-bullying* ada 78 data.



**Gambar 4.25** *Confusion matrix split data 60:40*

Dari **Gambar 4.25** didapatkan hasil *train* yang sebenarnya *non-bullying* dan terprediksi benar ada 300 data, yang sebenarnya *non-bullying* dan terprediksi salah ada 25 data, yang sebenarnya *bullying* dan terprediksi salah ada 22 data, yang sebenarnya *bullying* dan terprediksi benar ada 303 data. Pada *test* yang sebenarnya *bullying* dan terprediksi benar ada 108 data, yang sebenarnya *bullying* dan terprediksi salah ada 22 data, yang sebenarnya *non-bullying* dan terprediksi salah ada 23 data, yang sebenarnya *non-bullying* dan terprediksi benar ada 107 data.

#### 4.10 Evaluasi Model

Setelah melakukan proses *confusion matrix* dengan hasil yang didapatkan tersebut akan dihitung evaluasi model yang terdiri dari *precision*, *recall*, *f1-measure*, dan *accuracy* menggunakan metode *Support Vector Machine* (SVM). Untuk model split yang dipakai adalah 90:10, 80:20, 70: 30, dan 60:40. Nilainya tercantum pada **Tabel 4.3** dibawah ini:

**Tabel 4.3** Perbandingan *precision*, *recall*, *f1-measure*, dan *accuracy*

Model	Precision		Recall		F1-Measure		Accuracy
	Non-Bullying	Bullying	Non-Bullying	Bullying	Non-Bullying	Bullying	
SVM (90:10)	85%	88%	88%	85%	86%	86%	86%
SVM	78%	79%	80%	77%	79%	78%	78%

(80:20)							
SVM (70:30)	82%	81%	80%	83%	81%	82%	82%
SVM (60:40)	83%	82%	82%	83%	83%	83%	83%

Pada table diatas, menunjukan hasil *classification report* yang dibagi menjadi 2 kategori yaitu *non-bullying* dan *bullying* pada hasil *precision*, *recall*, *f1-measure*, dan *accuracy* menerapkan metode *Support Vector Machine* (SVM). Dengan masing-masing *split data* didapatkan hasil:

- a. Dengan *split data* 90:10 (90% *train data* 10% *test data*), hasil skor *precision* kategori *non-bullying* sebesar 85%, *bullying* sebesar 88%, skor *recall* kategori *non-bullying* sebesar 88%, *bullying* sebesar 85%, skor *f1-measure* kategori *non-bullying* sebesar 86%, *bullying* sebesar 86%, dan untuk skor *accuracy* sebesar 86%.
- b. Dengan *split data* 80:20 (80% *train data*, 20% *test data*), hasil skor *precision* kategori *non-bullying* sebesar 78%, *bullying* sebesar 79%, skor *recall* kategori *non-bullying* sebesar 80%, *bullying* sebesar 77%, skor *f1-measure* kategori *non-bullying* sebesar 79%, *bullying* sebesar 78%, dan untuk skor *accuracy* sebesar 78%.
- c. Dengan *split data* 70:30 (70% *train data*, 30% *test data*), hasil skor *precision* kategori *non-bullying* sebesar 82%, *bullying* sebesar 81%, skor *recall* kategori *non-bullying* sebesar 80%, *bullying* sebesar 83%, skor *f1-measure* kategori *non-bullying* sebesar 81%, *bullying* sebesar 82%, dan untuk skor *accuracy* sebesar 82%.
- d. Dengan *split data* 60:40 (60% *train data*, 40% *test data*), hasil skor *precision* kategori *non-bullying* sebesar 83%, *bullying* sebesar 82%, skor *recall* kategori *non-bullying* sebesar 82%, *bullying* sebesar 83%, skor *f1-measure* kategori *non-bullying* sebesar 83%, *bullying* sebesar 83%, dan untuk skor *accuracy* sebesar 83%.

## BAB V

### KESIMPULAN

Pada bab ini berisikan ringkasan dari masing-masing sub-sub bab dan mengasih saran supaya berguna dalam pengembangan penelitian ini selanjutnya.

Dari pengujian yang dilakukan pada BAB IV, maka hasil analisis sentiment komentar *cyberbullying* pada Instagram menggunakan metode *Support Vector Machine* (SVM) dengan masing-masing *split data* sebesar 90:10, 80:20, 70:30, 60:40 didapatkan skor *accuracy* 86%, 78%, 82%, 83%. Dari 4 perbandingan *split data* diatas, maka skor *accuracy* tertinggi yaitu sebesar 86%.

Saran untuk analisis sentiment komentar *cyberbullying* selanjutnya yaitu agar memperoleh hasil *accuracy* lebih maksimal dengan menggunakan metode *fiture selection* yang lain dan menambah jumlah data untuk proses analisis sentiment.

