

TESIS

**PERBANDINGAN KERNEL PADA METODE SUPPORT VECTOR
MACHINE (SVM) TERHADAP ANALISIS SENTIMEN
CYBERBULLYING**



Disusun oleh:

Nama : Sal Sabila Wijayanti
NIM : 21.51.2102
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2023

TESIS

**PERBANDINGAN KERNEL PADA METODE SUPPORT VECTOR MACHINE
(SVM) TERHADAP ANALISIS SENTIMEN CYBERBULLYING**

**COMPARISON OF KERNELS ON SUPPORT VECTOR MACHINE (SVM)
METHODS FOR ANALYSIS OF CYBERBULLYING SENTIMENT**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Sal Sablla Wijayanti
NIM : 21.51.2102
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2023

HALAMAN PENGESAHAN

**PERBANDINGAN KERNEL PADA METODE SUPPORT VECTOR MACHINE
(SVM) TERHADAP ANALISIS SENTIMEN CYBERBULLYING**

**COMPARISON OF KERNELS ON SUPPORT VECTOR MACHINE (SVM)
METHODS FOR ANALYSIS OF CYBERBULLYING SENTIMENT**

Dipersiapkan dan Disusun oleh

Sal Sabila Wijayanti

21.51.2102

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 01 Agustus 2023

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 01 Agustus 2023

Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

PERBANDINGAN KERNEL PADA METODE SUPPORT VECTOR MACHINE (SVM) TERHADAP ANALISIS SENTIMEN CYBERBULLYING

COMPARISON OF KERNELS ON SUPPORT VECTOR MACHINE (SVM) METHODS FOR ANALYSIS OF CYBERBULLYING SENTIMENT

Dipersiapkan dan Disusun oleh

Sal Sabila Wijayanti

21.51.2102

Telah Dimjikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, tanggal 01 Agustus 2023

Pembimbing Utama

Anggota Tim Penguji

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Alva Hendi M. S.T., M.Eng., Ph.D.
NIK. 190302493

Pembimbing Pendamping

M. Hanafi, S.Kom., M.Eng., Ph.D.
NIK. 190302024

Ainul Yaqin, M.Kom.
NIK. 1903022555

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 01 Agustus 2023
Direktur Program Pascasarjana

Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

DALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertanda tangan di bawah ini,

Nama mahasiswa : **Sul Sulaha Wijayanti**
NIM : **21512102**
Konsentrasi : **Manajemen Intelligensi**

Mencantumkan bahwa, Tesis adalah hasil buatan

Perencanaan dan Riset Pada Metode Support Vector Machine (SVM) Terhadap Analisis Sentimen e-Chatting

Dipada Pembimbing Utama : **Prof. Dr. Ema Dhami, S.Pd., M.Pd.**

Dipada Pembimbing Pendamping : **Azzul Yaqin, M.Kom.**

1. Saya telah dan sedang mengerjakan ASLI dan BUKAN PENCADAN tersebut untuk mendapatkan gelar akademis, baik di Universitas AN-NHARA Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Saya telah dan sedang mengerjakan penelitian ini secara mandiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing.
3. Untuk karya tulis ini telah terdapat karya saya sendiri yang lain, kecuali semua terdapat dengan cara dicantumkan sebagai catatan dalam hal-hal yang diperbolehkan dalam pengantar dan diperbolehkan dalam Daftar Pustaka pada karya tulis ini.
4. Penelitian ini tidak pernah dipublikasikan dalam bentuk lain (jurnal, prosiding, seminar, atau SAKS), tidak terdapat terbitan Universitas AN-NHARA Yogyakarta.
5. Pernyataan ini SAKS, tidak dengan persetujuan, sepihak, dan kesediaan dari lembaga yang menyetujui atau melaksanakan dengan persetujuan ini, maka SAYA bersedia menyetujui SANKSI ADMINISTRATIF dengan pemrosesan guru yang sudah ditetapkan, serta bersedia menerima sanksi yang akan diterima yang berlaku di Perguruan Tinggi.

Yogyakarta, 01 Agustus 2023
Yang Menandatangani,



Sul Sulaha Wijayanti

HALAMAN PERSEMBAHAN

Puji syukur kepada Allah SWT, yang Maha Esa, Dzat yang Maha Agung dan Maha Pengasih lagi Maha Penyayang, Tuhan semesta raya. Shalawat dan salam kepada baginda Rasulullah SAW, yang telah mengenalkan Tuhan Yang Maha Esa sebagai kebenaran sejati pada jiwa-jiwa pencinta-Nya, untuk jiwa-jiwa suci yang senantiasa menghembuskan nafas-Nya, menuliskan keberkahan nama-Nya. Dengan segala rasa cinta dan kasih tesis ini saya persembahkan kepada:

1. Allah SWT, atas segala nikmat dan karunia-Nya sehingga tesis ini dapat dibuat dengan baik dan dapat diselesaikan dengan lancar.
2. Segenap keluarga saya yang senantiasa mendoakan, mendukung, memberi semangat, nasihat, serta menyayangi dengan sepenuh hati.
3. Dosen pembimbing yang selalu memberikan arahan dan motivasi selama pengerjaan tesis.
4. Bapak/Ibu Dosen atas segala ilmu yang telah diberikan.
5. Teman-teman kelas Magister Teknik Informatika Angkatan 27 A yang telah menjadi bagian dalam proses belajar dan semoga kita senantiasa diberikan kesehatan dan kesuksesan.

Serta semua pihak yang membantu serta mendukung saya yang tidak bisa saya sebutkan satu persatu.

HALAMAN MOTTO

"Allah-lah yang menciptakan tujuh langit dan seperti itu pula bumi. Perintah Allah berlaku padanya, agar kamu mengetahui bahwasanya Allah Maha Kuasa atas segala sesuatu, dan sesungguhnya Allah ilmu-Nya benar-benar meliputi segala sesuatu." (QS. At-Thalaq: 12)

"Siapa yang menempuh jalan untuk mencari ilmu, maka Allah akan memudahkan baginya jalan menuju surga." (HR. Muslim, no. 2699)

"Maka sesungguhnya bersama kesulitan ada kemudahan. Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari sesuatu urusan), tetaplah bekerja keras (untuk urusan yang lain). Dan hanya kepada Tuhanmulah engkau berharap." (QS. Al-Insyirah: 6-8)

Jawaban dari sebuah keberhasilan adalah terus mencoba belajar tanpa putus asa.



KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat, hidayah, serta karuni-Nya, sehingga penulis dapat menyelesaikan tesis ini dengan lancar. Tidak lupa sholawat serta salam penulis haturkan kepada Nabi Muhammad SAW. Tesis ini disusun untuk memenuhi salah satu syarat menempuh jenjang Program Pascasarjana Universitas Amikom Yogyakarta.

Seiring dengan terselesaikannya penyusunan tesis ini, tak lupa penulis menyampaikan terima kasih dan penghargaan setinggi-tingginya kepada semua pihak yang telah membantu memberikan arahan, bimbingan dan petunjuk serta motivasi dalam proses penyusunannya, yakni ucapan terimakasih kepada:

1. Prof. Dr. M. Suyanto, M.M, selaku Rektor Universitas AMIKOM Yogyakarta.
2. Prof. Dr. Kusri, M.Kom, selaku Direktur Program Pascasarjana.
3. Prof. Dr. Ema Utami, S.Si., M.Kom, selaku dosen pembimbing utama yang telah memberikan arahan, bimbingan, dan nasihat dalam proses penyusunan tesis ini.
4. Bapak Ainul Yaqin, M.Kom, selaku dosen pembimbing pendamping yang telah membantu penulis dengan memberikan masukan-masukan dan arahan kepada penulis.
5. Keluarga penulis yang telah memberikan dukungan, doa, nasihat, dan motivasi kepada penulis.
6. Bapak/Ibu Dosen dan staff Universitas Amikom Yogyakarta yang telah banyak memberikan ilmu dan fasilitas selama penulis berkuliah.
7. Teman-teman kelas Magister Teknik Informatika Angkatan 27 A atas segala bantuan, doa, dan dukungan semangatnya.
8. Semua pihak yang telah membantu dalam penyelesaian tesis ini yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa tesis ini masih terdapat kekurangan, maka dari itu penulis berharap kepada semua pihak agar dapat menyampaikan kritik dan saran untuk menambah kesempurnaan tesis ini. Sekian dari penulis, apabila terdapat kesalahan dan kekurangan mohon maaf sebesar-besarnya.

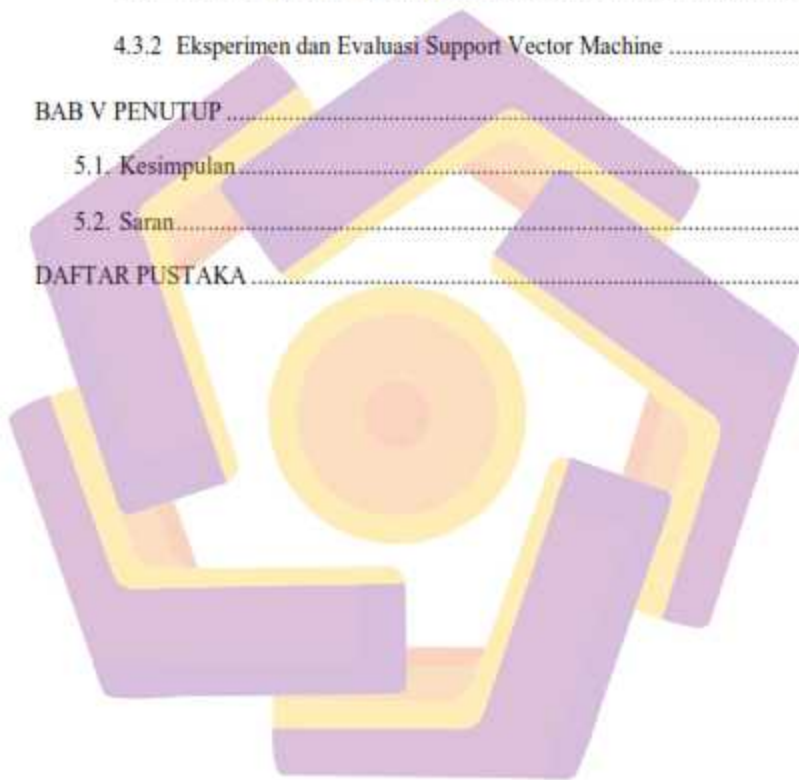
Yogyakarta, 01 Agustus 2023
Penulis

DAFTAR ISI

JUDUL	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSETUJUAN	iv
HALAMAN PERNYATAAN KEASLIAN TESIS	v
HALAMAN PERSEMBAHAN	vi
HALAMAN MOTTO	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR ISTILAH	xvi
INTISARI	xvii
<i>ABSTRACT</i>	xviii
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	4
1.3. Batasan Masalah	4
1.4. Tujuan Penelitian	5
1.5. Manfaat Penelitian	6
BAB II TINJAUAN PUSTAKA	7
2.1. Tinjauan Pustaka	7
2.2. Keaslian Penelitian	11

2.3. Landasan Teori.....	15
2.3.1 Machine Learning.....	15
2.3.2 Data Mining.....	16
2.3.3 Text Mining.....	18
2.3.4 Analisis Sentimen.....	18
2.3.5 Support Vector Machine (SVM).....	19
2.3.6 TF-IDF.....	26
2.3.7 Confusion Matrix.....	27
BAB III METODE PENELITIAN.....	29
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	29
3.2. Metode Pengumpulan Data.....	29
3.3. Metode Analisis Data.....	30
3.4. Alur Penelitian.....	31
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	34
4.1 Tahap Crawling Tweet.....	34
4.2 Preprocessing Data.....	35
4.2.1 Handle Duplicates.....	36
4.2.2 Pelabelan Data.....	37
4.2.3 Cleansing.....	37
4.2.4 Tokenizing.....	38
4.2.5 Stopword Removal.....	38

4.2.6 Case Folding.....	39
4.2.7 Stemming.....	39
4.3 Klasifikasi Support Vector Machine.....	39
4.3.1 Pembagian Data Latih dan Data Uji.....	39
4.3.2 Eksperimen dan Evaluasi Support Vector Machine.....	40
BAB V PENUTUP.....	75
5.1. Kesimpulan.....	75
5.2. Saran.....	76
DAFTAR PUSTAKA.....	77



DAFTAR TABEL

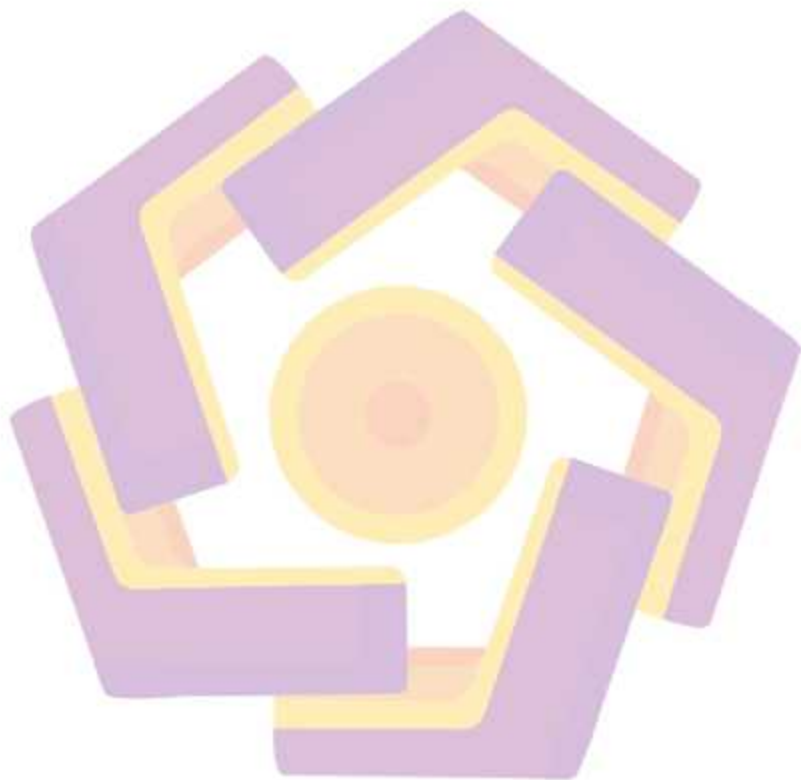
Tabel 2. 1. Matriks literatur review dan posisi penelitian.....	11
Tabel 2. 2. <i>Confusion Matrix</i>	27
Tabel 4. 1. Contoh Pelabelan Dataset.....	37
Tabel 4. 2. Pembagian Data.....	40
Tabel 4. 3. Perangkat Eksperimen.....	40
Tabel 4. 4. Penentuan Degree.....	46
Tabel 4. 5 Penentuan C.....	48
Tabel 4. 6. Penentuan Gamma RBF.....	51
Tabel 4. 7 Penentuan Gamma Sigmoid.....	52
Tabel 4. 8. Penentuan Gamma Polynomial.....	53
Tabel 4. 9. Nilai Gamma yang Dipilih.....	54
Tabel 4. 10. Hasil SVM Kernel Linear.....	56
Tabel 4. 11. Hasil SVM Kernel Polynomial.....	60
Tabel 4. 12. Hasil SVM Kernel RBF.....	65
Tabel 4.13. Hasil SVM Kernel Sigmoid.....	68
Tabel 4. 14. Perbandingan Kernel SVM.....	72
Tabel 4. 15. Uji Coba Pengaruh Parameter.....	73

DAFTAR GAMBAR

Gambar 2. 1. Irisan Pola <i>Data Mining</i>	17
Gambar 2. 2. Support Vector Machine (SVM).....	20
Gambar 2. 3. Kernel SVM.....	21
Gambar 2. 4. Parameter C.....	24
Gambar 2. 5. Parameter Gamma (Tinggi).....	25
Gambar 2. 6. Parameter Gamma (Rendah).....	25
Gambar 3. 1. Alur Pengumpulan Data.....	30
Gambar 3. 2. Alur Penelitian.....	33
Gambar 4. 1. Hasil Crawling.....	34
Gambar 4. 2. Tampilan Dataset.....	35
Gambar 4. 3. Handle Duplicates.....	36
Gambar 4. 4. Hasil Cleansing.....	38
Gambar 4. 5. Tokenizing.....	38
Gambar 4. 6. Script Stopword Removal.....	39
Gambar 4. 7. Skenario 1 TF-IDF.....	41
Gambar 4. 8. Hasil TF-IDF Skenario 80:20.....	42
Gambar 4. 9. Hasil TF-IDF Skenario 70:30.....	43
Gambar 4. 10. Hasil TF-IDF Skenario 60:40.....	43
Gambar 4. 11. Pengujian Nilai Degree.....	45
Gambar 4. 12. Hasil Pengujian Nilai degree.....	46

Gambar 4. 13. Pengujian Nilai C.....	47
Gambar 4. 14. Hasil Pengujian Nilai C.....	48
Gambar 4. 15. Pengujian Nilai Gamma RBF.....	49
Gambar 4. 16. Pengujian Nilai Gamma Sigmoid.....	50
Gambar 4. 17. Pengujian Nilai Gamma Polynomial.....	50
Gambar 4. 18. Hasil Pengujian Gamma RBF.....	51
Gambar 4. 19. Hasil Pengujian Gamma Sigmoid.....	52
Gambar 4. 20. Hasil Pengujian Gamma Polynomial.....	53
Gambar 4. 21. Pseudocode Kernel Linear.....	56
Gambar 4. 22. Decision Boundary Linear Skenario 1.....	57
Gambar 4. 23. Decision Boundary Linear Skenario 2.....	58
Gambar 4. 24. Decision Boundary Linear Skenario 3.....	59
Gambar 4. 25. Pseudocode Kernel Polynomial.....	60
Gambar 4. 26. Decision Boundary Polynomial Skenario 1.....	61
Gambar 4. 27. Decision Boundary Polynomial Skenario 2.....	62
Gambar 4. 28. Decision Boundary Polynomial Skenario 3.....	63
Gambar 4. 29. Pseudocode Kernel RBF.....	64
Gambar 4. 30. Decision Boundary RBF Skenario 1.....	65
Gambar 4. 31. Decision Boundary RBF Skenario 2.....	66
Gambar 4. 32. Decision Boundary RBF Skenario 3.....	67
Gambar 4. 33. Pseudocode Kernel Sigmoid.....	68
Gambar 4. 34. Decision Boundary Sigmoid Skenario 1.....	69
Gambar 4. 35. Decision Boundary Sigmoid Skenario 2.....	70

Gambar 4. 36. Decision Boundary Sigmoid Skenario 3 71



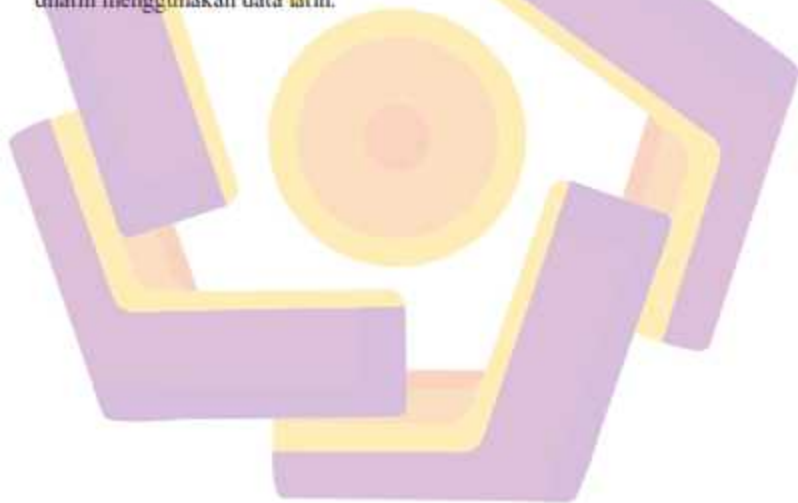
DAFTAR ISTILAH

Dataset: kumpulan data yang terorganisir yang mencakup informasi atau pengamatan tentang suatu topik atau domain tertentu.

Kernel: metode sederhana yang digunakan untuk memetakan dan mengubah data nonlinier berdimensi rendah menjadi ruang berdimensi lebih tinggi.

Data latih: disebut juga sebagai dataset pelatihan (*training dataset*) merupakan input yang diberikan ke algoritma pembelajaran mesin untuk mempelajari pola atau hubungan antara fitur (atribut) input dengan target output yang sesuai.

Data uji: disebut juga sebagai dataset uji (*test dataset*), merupakan sebagian data dari keseluruhan dataset yang digunakan untuk menguji kinerja model yang telah dilatih menggunakan data latih.



INTISARI

Saat ini media sosial telah berkembang pesat dan sudah menjadi bagian rutinitas sehari-hari. Seiring dengan berkembangnya media sosial terdapat berbagai penyalahgunaan dan penyimpangan terhadap etika berinteraksi yang dilakukan penggunanya. Salah satu bentuk penyalahgunaan media sosial yaitu perundungan (*cyberbullying*). Akhir-akhir *cyberbullying* semakin sering terjadi dan kejadian tersebut masih sulit terdeteksi. Oleh karena itu, peneliti melakukan analisis terhadap *cyberbullying* di media sosial Twitter menggunakan *Support Vector Machine* (SVM).

Data yang digunakan merupakan data hasil *crawling* Twitter. Data ini termasuk kedalam data *non-linear* sehingga memerlukan kernel untuk melakukan proses *text mining*. Namun, sampai saat ini belum ada penelitian spesifik tentang kernel apa yang baik untuk digunakan pada kasus *cyberbullying*. Oleh karena itu, peneliti melakukan percobaan untuk mencari kernel yang terbaik diantara empat kernel yaitu Polynomial, Radial Basis Function (RBF), Sigmoid, dan Linear.

Dari hasil percobaan yang dilakukan, peneliti menetapkan kernel terbaik berdasarkan nilai akurasi. Pengujian terhadap empat jenis kernel yang telah dilakukan, diketahui bahwa kernel Polynomial memiliki akurasi terbaik. Pada skenario 1 kernel polynomial mendapat akurasi 86,49%. Pada skenario 2 kernel polynomial mendapat akurasi 86,12%. Pada skenario 3 kernel polynomial mendapat akurasi 82,70%.

Kata kunci: *cyberbullying*, SVM, kernel, akurasi.

ABSTRACT

Nowadays, social media has grown rapidly and has become part of our daily routine. Along with the development of social media, there are various abuses and deviations from the ethics of interaction by its users. One form of abuse of social media is bullying (cyberbullying). Lately cyberbullying has become more frequent and the incident is still difficult to detect. Therefore, researchers conducted an analysis of cyberbullying on social media Twitter using the Support Vector Machine (SVM).

The data used is the result of Twitter crawling. This data is included in non-linear data so it requires a kernel to carry out the text mining process. However, until now there has been no specific research on what kernel is good for use in cases of cyberbullying. Therefore, researchers conducted an experiment to find the best kernel among the four kernels, namely Polynomial, Radial Basis Function (RBF), Sigmoid, and Linear.

From the results of the experiments conducted, the researchers determined the best kernel based on the accuracy value. Testing on the four types of kernels that have been carried out, it is known that the Polynomial kernel has the best accuracy. In scenario 1 the polynomial kernel gets 86.49% accuracy. In scenario 2 the polynomial kernel gets an accuracy of 86.12%. In scenario 3 the polynomial kernel gets 82.70% accuracy.

Keyword: cyberbullying, SVM, kernel, accuracy.

BAB I

PENDAHULUAN

1.1.Latar Belakang Masalah

Perkembangan media sosial yang sangat pesat pada saat ini mendorong masyarakat untuk tanggap terhadap etika berinteraksi atau berkomunikasi di dunia maya. Media sosial memberikan fasilitas untuk pengguna dalam menjalin hubungan dan ikatan sosial. Melalui platform media sosial memungkinkan pengguna untuk saling bertukar informasi dan menyebarkannya dengan mudah. Sehingga fenomena berinteraksi di media sosial berpengaruh terhadap gaya hidup, cara pandang, dan perilaku masyarakat.

Salah satu jenis media sosial yang sering digunakan saat ini yaitu *Twitter*. *Twitter* merupakan media sosial untuk berbagi pesan atau foto. *Twitter* telah digunakan oleh berbagai kalangan masyarakat, seperti selebritis, pejabat, pelajar, mahasiswa, hingga masyarakat umum. Namun, kemudahan dan keseruan yang didapat dari berinteraksi di media sosial tidak lepas dari penyalahgunaan oleh penggunanya. Salah satu bentuk penyalahgunaan media sosial *Twitter* yaitu perundungan di dunia maya atau *cyberbullying*.

Menurut UNICEF, *cyberbullying* atau perundungan di dunia maya merupakan tindakan *bullying* atau perundungan yang dilakukan dengan memanfaatkan teknologi digital. *Cyberbullying* bertujuan untuk menakuti korban, memancing kemarahan, atau mempermalukan korban. Perbedaan *bullying* langsung dan *cyberbullying* yaitu pada kasus *cyberbullying* meninggalkan jejak

digital. Jejak digital dapat berupa rekaman atau catatan yang dapat digunakan sebagai bukti untuk membantu menghentikan perbuatan tersebut.

Perundungan di dunia maya (*cyberbullying*) semakin sering terjadi. Akan tetapi kasus perundungan masih sulit di deteksi. Oleh karena itu, klasifikasi *cyberbullying* sangat diperlukan untuk mempermudah mendeteksi suatu kalimat yang mengandung unsur perundungan atau tidak. Salah satu cara mengklasifikasi *cyberbullying* adalah dengan pendekatan pembelajaran mesin atau *machine learning* yaitu dengan analisis sentimen.

Analisis sentimen merupakan salah satu rumpun ilmu pada bidang *Natural Language Processing* (NLP) yang bertujuan untuk mengetahui perasaan yang ada pada sebuah teks. Analisis sentimen adalah serangkaian tahapan untuk memahami, mengolah, dan mengekstrak suatu data teks untuk memperoleh informasi polaritas sentimen dalam suatu kalimat. Polaritas sentimen dapat berupa sentimen positif, sentimen negatif, dan sentimen netral. Akan tetapi, pada penelitian ini polaritas sentimen akan dibagi menjadi sentimen *cyberbullying* dan sentimen *non-cyberbullying*.

Penelitian tentang analisis *cyberbullying* telah banyak diteliti oleh peneliti sebelumnya, seperti analisis *cyberbullying* di media sosial twitter tentang *fans Manchester United* (Dimas Bagus Aditya, 2022). Hasil penelitian ini menunjukkan bahwa bentuk *cyberbullying* paling banyak ditemukan dalam kosa kata kasar. Sedangkan (Anisa Putri, dkk, 2022) dalam penelitiannya tentang analisis *cyberbullying* terhadap KPOP di twitter menghasilkan lebih banyak sentimen netral

dibandingkan dengan sentimen negatif yang berisi ketakutan, *bullying*, sedih, kecewa, dan unsur protes.

Dalam penelitian ini, penulis akan menggunakan metode *Support Vector Machine* (SVM). Penelitian menggunakan metode SVM pernah dilakukan oleh Muh.Fitra (2021) untuk mengklasifikasi tweet yang mengandung *cyberbullying* dan *non-cyberbullying*. Penelitian menggunakan 100 data dengan *confusion matrix* mendapat nilai *recall* 64%, *precision* 58%, dan *accuracy* 70%. Penelitian *cyberbullying* menggunakan SVM juga pernah dikolaborasi dengan metode *Naive Bayes* (Tosin Ige, 2022). Penelitian yang dilakukan Tosin (2022) berhasil mendapat nilai akurasi 92% dari eksperimen yang dilakukan.

Penulis memilih metode SVM karena metode ini berakar pada teknik pembelajaran statistik yang dapat memberikan hasil yang lebih baik dibandingkan dengan metode lainnya. Kinerja SVM diawali dengan pendekatan kernel untuk fitur awal pada himpunan data (Muh.Fitra, 2021). Menurut Dewi Fortuna (2022) algoritma SVM merupakan metode yang umum digunakan untuk klasifikasi teks karena lebih tidak rentan terhadap *overfitting* jika dibandingkan dengan metode yang lain. Selain itu SVM dapat digunakan untuk mengolah data yang memiliki dimensi tinggi dengan cara memanfaatkan ruang kernel. Terdapat berbagai macam jenis kernel SVM, seperti linear kernel, polynomial kernel, RBF kernel, dan sigmoid kernel. Akan tetapi, belum ada kepastian tentang kernel mana yang paling baik dan kernel yang buruk untuk digunakan pada kasus tertentu (Thalita Meisya Permata Aulia, dkk, 2021). Oleh karena itu, penulis akan melakukan penelitian tentang *cyberbullying* dengan menggunakan SVM dan melakukan perbandingan

kernel untuk mengetahui kernel mana yang paling akurat untuk melakukan klasifikasi pada kasus data *cyberbullying*.

1.2. Rumusan Masalah

Berdasarkan uraian latar belakang masalah diatas, maka permasalahan yang akan dibahas pada penelitian ini dirumuskan sebagai berikut:

- a. Bagaimana mengimplementasikan SVM untuk klasifikasi tweet *cyberbullying*?
- b. Berapakah nilai akurasi dari setiap skenario pada kernel linear, polynomial, RBF, dan sigmoid?
- c. Kernel SVM manakah yang memiliki akurasi tertinggi?

1.3. Batasan Masalah

Batasan masalah pada penelitian ini ditentukan untuk mengerucutkan pembahasan. Batasan masalah yang dilakukan adalah sebagai berikut:

- a. Data yang digunakan merupakan data dari proses crawling di twitter menggunakan kata kunci pemicu *cyberbullying*.
- b. Data yang digunakan adalah *tweet* berbahasa Indonesia.
- c. Data hasil *crawling* akan dipilih 50 *tweet* untuk setiap kata kunci yang digunakan.
- d. Penentuan kata kunci berdasarkan wawancara peneliti terdahulu dengan ahli Bahasa (Christevan Destisus, dkk, 2020).
- e. Kata kunci (Christevan Destisus, dkk, 2020) yang digunakan pada penelitian ini hanya kata kunci yang berbahasa Indonesia.

- f. Pelabelan data dilakukan berdasarkan ilmu psikologi yang dikerjakan secara manual oleh psikolog.
- g. Data *tweet* dikelompokkan menjadi *tweet cyberbullying* dan *non-cyberbullying*.
- h. Metode yang digunakan adalah SVM.
- i. Penelitian ini menggunakan *platform* Google Colaboratory.
- j. *Tools* atau *library* yang digunakan yaitu *twint*, *pandas*, *nest_asyncio*, *numpy*, *nlk*, *sastrawi*, *sklearn*, *matplotlib*.
- k. Jenis Kernel SVM yang dibandingkan yaitu *Polynomial*, *Radial Basic Function (RBF)*, *Sigmoid* dan *Linear*.
- l. Pengujian dilakukan menggunakan *confusion matrix*.
- m. Tahap penelitian ini sampai pada membandingkan performa jenis Kernel SVM dan menentukan kernel yang memiliki performa terbaik berdasarkan pengujian *confusion matrix*.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dijelaskan, maka tujuan dalam penelitian ini adalah sebagai berikut:

- a. Mengetahui *tweet* yang mengandung unsur *cyberbullying*.
- b. Menerapkan empat jenis Kernel metode SVM untuk menganalisis *cyberbullying*.
- c. Mengetahui performa dari setiap Kernel SVM yang digunakan melalui pengujian *confusion matrix*.
- d. Mengetahui performa tertinggi Kernel SVM dalam melakukan analisis sentimen *cyberbullying*

1.5. Manfaat Penelitian

Manfaat dari hasil penelitian ini yang diharapkan oleh penulis adalah sebagai berikut:

- a. Secara teoritis, berdasarkan tujuan penelitian ini maka diharapkan dapat mempunyai manfaat menambah pengetahuan di bidang *Natural Language Processing*.
- b. Secara praktis, penelitian ini diharapkan dapat memiliki manfaat sebagai berikut:
 - 1) Dapat mengetahui jenis Kernel SVM yang memiliki akurasi tertinggi untuk melakukan analisis sentimen *cyberbullying*.
 - 2) Dapat mengetahui kalimat yang termasuk *cyberbullying* atau *non-cyberbullying*.



BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Terdapat penelitian yang membahas tentang analisis bentuk cyberbullying yang dilakukan pengguna twitter terhadap *fans* Manchester United (Dimas Bagus Aditya, 2022). Kelebihan dari penelitian ini yaitu terdapat detail penjelasan mengenai bentuk-bentuk cyberbullying dari hasil analisis yang dilakukan. Selain itu dalam penelitian ini diuraikan pembahasan data tweet dari beberapa akun yang digunakan sebagai dataset. Hasil dari penelitian ini menunjukkan bahwa kasus *cyberbullying* paling banyak ditemukan dalam bentuk penggunaan kosa kata kasar. Dalam penelitian ini menyarankan untuk penelitian selanjutnya agar menggunakan data dengan kasus akun tertentu.

Penelitian lain tentang *cyberbullying* juga pernah dilakukan untuk menganalisis pengguna twitter terhadap KPOP (Anisa Putri, dkk, 2022). Pada penelitian ini digunakan metode *Naive Bayes* untuk mengklasifikasi sentimen tweet positif, negatif, netral, dan mengetahui *trend bullying*. Kelebihan dari penelitian ini yaitu terdapat pembahasan lengkap terkait data yang digunakan sampai pada kendala yang dialami. Namun pada penelitian ini data yang digunakan lebih banyak mengandung sentimen netral. Sehingga peneliti dalam penelitian ini menyarankan menggunakan algoritma yang berbeda dan memperbanyak data latih agar hasil performa yang didapat lebih akurat.

Muh. Fitra, dkk (2021) pernah melakukan penelitian tentang analisis sentimen di twitter menggunakan SVM untuk mengklasifikasi tweet yang

mengandung unsur *cyberbullying* dan *non-cyberbullying*. Kelebihan pada penelitian ini yaitu data disajikan secara spesifik. Akan tetapi penelitian ini masih menggunakan data latih dan data uji yang berjumlah sedikit (100 data). Oleh karena itu, peneliti dalam penelitian ini menyarankan untuk menambah data *training* dan meningkatkan kinerja SVM.

Chistevan Destisus, dkk, (2020) juga pernah melakukan penelitian tentang *cyberbullying* menggunakan data tweet. Peneliti membandingkan metode dengan metode Information Gain. Hasil dari penelitian ini yaitu metode SVM mendapat akurasi 80%, precision 75,1%, recall 96%, dan f-measure 85%. Sedangkan metode Information Gain mendapat akurasi 86%, precision 81%, recall 95%, dan f-measure 87%. Kelebihan pada penelitian ini menjelaskan secara rinci tentang teknik pengolahan data. Namun pada penelitian ini hanya menerapkan satu jenis kernel yaitu polynomial.

Penelitian *cyberbullying* pernah dilakukan dengan menerapkan algoritma *machine learning* yaitu *Multinomial Naive Bayes* dan optimasi SVM menggunakan dataset twitter, kaggle, wikipedia, dan youtube (Tosin Ige, dkk, 2022). Peneliti melakukan pengembangan *chatbot* untuk mendeteksi dan menghalangi pesan intimidasi atau membuat filter untuk pesan yang masuk dan keluar. Sehingga pesan tidak sampai ke penerima jika mengandung unsur intimidasi. Kelebihan pada penelitian ini berhasil mengimplementasikan metode dengan akurasi 92% dan berhasil mendeteksi berbagai jenis *cyberbullying*. Jenis *cyberbullying* yang terdeteksi pada sistem yang dikembangkan yaitu rasisme,

ujaran kebencian, agresi, penghinaan, dan *toxic*. Namun, penelitian ini memiliki kelemahan pada keterbatasan dataset.

Penelitian mengenai *cyberbullying* di twitter tidak hanya pada kasus kalimat berbahasa Indonesia, akan tetapi juga pernah dilakukan penelitian pada sentimen bahasa Arab (Samar Almutiry, dkk, 2021). Pada penelitian ini peneliti memiliki tujuan untuk mencari alat penambangan data terbaik dengan cara membandingkan WEKA dengan Python. Kelebihan pada penelitian ini menjelaskan detail tentang hasil perbandingan yang dilakukan peneliti. Hasilnya yaitu dalam hal klasifikasi WEKA lebih baik daripada Python, namun dalam hal waktu yang dibutuhkan untuk membangun model maka Python membutuhkan lebih sedikit waktu dibandingkan WEKA. Akan tetapi pelabelan data hanya dilakukan dengan sistem tanpa melibatkan pakar.

Penelitian lain tentang *cyberbullying* pernah dilakukan untuk menganalisis komentar Instagram menggunakan metode SVM dan menggabungkan metode *Semantic Similarity* (Lintani Alfina Hajar Raudhoti, dkk, 2020). Kelebihan pada penelitian ini menerapkan *semantic similarity* untuk mengumpulkan kata-kata serupa lalu dibuat kamus sehingga dapat menggantikan kata yang tidak dikenali dalam data. Pada penelitian ini dijelaskan bahwa kualitas kamus berpengaruh pada hasil pengujian. Kelemahan penelitian ini belum bisa mendeteksi kata yang memiliki ejaan berbeda atau kesalahan ejaan. Sehingga peneliti menyarankan untuk meningkatkan performa kamus dan melakukan normalisasi pada tahap *preprocessing*.

Algoritma SVM dan TF-IDF juga pernah diterapkan untuk analisis sentimen *cyberbullying* (Wahyu Adi Prabowo, 2020). Kelebihan dalam penelitian ini adalah meningkatkan akurasi SVM dengan TF-IDF. Hasil dari eksperimen yang dilakukan peneliti mendapat akurasi 93%, *precision* 95% dan *recall* 97%. Akan tetapi pada penelitian ini tidak dijelaskan tentang topik dataset yang digunakan.

Dari kajian penelitian terdahulu, maka penulis akan melakukan penelitian untuk menganalisis *cyberbullying* menggunakan metode SVM. Dataset yang digunakan penulis merupakan data *tweet* yang telah diberikan label *cyberbullying* dan *non-cyberbullying*. Pada tahap analisis SVM penulis akan menggunakan empat jenis kernel SVM yaitu linear, RBF, polynomial, dan sigmoid. Peneliti akan mencari jenis kernel yang memiliki akurasi terbaik untuk diterapkan pada kasus *cyberbullying*. Hal ini dikarenakan pemilihan kernel SVM sangat berpengaruh terhadap performa hasil analisis.

Penelitian mengenai perbandingan kernel SVM pernah dilakukan dalam studi kasus sentimen vaksinasi Covid-19 (Thalita Meisya Permata Aulia, dkk, 2021). Hasil penelitian ini menunjukkan kernel yang memiliki akurasi lebih tinggi yaitu sigmoid dan linear. Dalam penelitian ini menyarankan mengganti perbandingan data latih dan dataset serta menambah data pada *slangword*.

Penelitian lain mengenai perbandingan kernel SVM juga pernah diterapkan untuk *tren marketplace* berdasarkan klasifikasi ulasan pelanggan (Dwi Latifah Rianti, dkk, 2021). Hasil dari penelitian ini menunjukkan kernel yang memiliki akurasi terbaik adalah sigmoid. Kelemahan pada penelitian ini karena hanya menerapkan satu skenario pengujian.

2.2. Keaslian Penelitian

Tabel 2. 1. Matriks literatur review dan posisi penelitian Perbandingan Kernel pada Metode Support Vector Machine (SVM) Terhadap Analisis Sentimen Cyberbullying

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Analisis Cyberbullying Terhadap Fans Manchester United di Twitter	Dimas Bagus-Aditya, OSF PREPRINTS, 2022	Menganalisis bentuk cyberbullying yang dilakukan warganet	Cyberbullying paling banyak ditemukan yaitu penggunaan kosa kata kasar	Menyarankan untuk penelitian dengan studi kasus akun tertentu dan penelitian lanjutan di luar cyberbullying twitter	Dataset didapat dari proses crawling di Twitter menggunakan kata kunci pemicu cyberbullying.
2	Analisis Sentimen Cyberbullying KPOP di Media Sosial Twitter Menggunakan Metode Naive Bayes	Anisa Putri, Ari Muzakir, Syntax Literate: Jurnal Ilmiah Indonesia, 2022	Melakukan klasifikasi berdasarkan sentimen tweet positif, negatif, netral dan mengetahui trend dalam tindakan bullying	Berhasil menemukan cyberbullying dengan menggunakan metode multinomial naïve bayes dan dievaluasi menggunakan confusion matrix	Menyarankan menggunakan algoritma yang berbeda dan memperbanyak data latih agar hasil performa lebih akurat	Metode yang akan digunakan untuk menganalisis cyberbullying adalah SVM

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Analisis Sentiment <i>Cyberbullying</i> pada Sosial Media Twitter Menggunakan Metode <i>Support Vector Machine</i>	Muh. Fitra Rizki, dkk., Jurnal Mahasiswa Teknik Informatika, 2021	Mengklasifikasi tweet yang mengandung unsur <i>cyberbullying</i> dan <i>non-cyberbullying</i>	Hasil evaluasi dari 100 data dengan <i>confusion matrix</i> mendapat nilai <i>recall</i> 64%, <i>precision</i> 58%, dan <i>accuracy</i> 70%	Menyarankan menambah data <i>training</i> dan meningkatkan kinerja SVM	Menggunakan tiga skenario pembagian data <i>training</i> dan data <i>testing</i> .
4	<i>Support Vector Machine VS Information Gain: Analisis Sentimen Cyberbullying</i> di Twitter Indonesia	Christevan Destisus, Wellia, Suryasari, ULTIMA InfoSys, 2020	Memberi hasil akurasi dari data tweet <i>cyberbullying</i> dengan SVM dan <i>Information Gain</i>	Metode SVM mendapat akurasi 80%, <i>precision</i> 75,1%, <i>recall</i> 96%, dan <i>f-measure</i> 85%. Sedangkan metode <i>Information Gain</i> mendapat akurasi 86%, <i>precision</i> 81%, <i>recall</i> 95%, dan <i>f-measure</i> 87%	Hanya menggunakan kernel <i>polynomial</i>	Metode yang digunakan hanya SVM akan tetapi menggunakan empat kernel dan membandingkan kernel
5	<i>AI Powered Anti-Cyber Bullying System using Machine Learning Algorithm of Multinomial Naïve Bayes and Optimized Linear Support Vector Machine</i>	Tosin Ige, Sikiru Adewale, IJACSA, 2022	Mengembangkan <i>chatbot</i> untuk mendeteksi dan menghalangi pesan intimidasi yang masuk dan keluar sehingga pesan tidak sampai ke penerima	Metode <i>Multinomial Naive Bayes</i> dan optimasi SVM berhasil diimplementasikan untuk mendeteksi <i>cyberbullying</i> dengan akurasi 92%. Jenis <i>cyberbullying</i> yang terdeteksi yaitu rasisme, ujaran kebencian, agresi, penghinaan, dan <i>toxic</i>	Keterbatasan model yang dikembangkan dan keterbatasan dataset. Dataset didapat dari kaggle, twitter, wikipedia, dan youtube	Penelitian menggunakan dataset dari proses <i>crawling tweet</i> . <i>Crawling</i> dilakukan dengan mencari tweet yang sesuai <i>keyword</i> yang telah ditentukan ahli bahasa.

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6	<i>Arabic CyberBullying Detection Using Arabic Sentiment Analysis</i>	Samar Almutiry, Mohamed Abdel Fattah, Egyptian Journal of Language Engineering, 2021	Mencari alat pembandingan data terbaik (WEKA dan Python) dalam mendeteksi cyberbullying berbahasa Arab di twitter	Dalam hal klasifikasi WEKA lebih baik daripada Python, namun dalam hal waktu yang dibutuhkan untuk membangun model maka Python membutuhkan lebih sedikit waktu dibandingkan WEKA	Pelabelan data dilakukan dengan sistem, tidak melibatkan pakar	Menggunakan python untuk <i>crawling</i> dan analisis, pelabelan dilakukan oleh psikolog
7	Identifikasi <i>Cyberbullying</i> pada Kolom Komentar Instagram dengan Metode <i>Support Vector Machine</i> dan <i>Semantic Similarity</i>	Lintani Afina Hajar Rindhoi, Anisa Herdiani, Ade Romadhony, J-COSINE, 2020	Menerapkan SVM untuk mengklasifikasi <i>cyberbullying</i> dan menggunakan <i>semantic similarity</i> untuk mengumpulkan kata-kata serupa untuk menggantikan kata yang tidak dikenali dalam dat	Kualitas kamus berpengaruh pada hasil pengujian	Menyarankan untuk meningkatkan performa kamus dan melakukan normalisasi pada tahap <i>preprocessing</i>	Dataset yang digunakan dari twitter.

Tabel 2.1 (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
8	<i>Sentiment Analysis for Detecting Cyberbullying Using TF-IDF and SVM</i>	Wahyu Adi Prabowo, Fitriani Azizah, RESTI JOURNAL, 2020	Meningkatkan akurasi SVM dengan TF-IDF dalam menganalisis <i>cyberbullying</i>	Algoritma SVM dapat diterapkan untuk menganalisis komentar <i>cyberbullying</i> dan <i>non-cyberbullying</i> pada orgram. Hasil yang diperoleh dari eksperimen yaitu mendapat akurasi 93%, precision 95% dan recall 97%.	Tidak dijelaskan topik sentimen yang digunakan pada dataset yang dapat memicu <i>cyberbullying</i>	Data yang digunakan merupakan hasil <i>crawling twitter</i> yang diperoleh dari <i>keyword</i> yang diduga menjadi pemicu <i>cyberbullying</i> .
9	Perbandingan Kernel <i>Support Vector Machine (SVM)</i> dalam Penerapan Analisis Sentimen Vaksinasi Covid-19	Thalita Meisya Permata Aulia, dkk., SINTECH, 2021	Melakukan analisis sentimen terhadap vaksin dan meneliti kernel apa yang baik untuk sentimen	Kernel yang memiliki nilai akurasi lebih tinggi yaitu Sigmoid dan Linear	Menyarankan mengganti perbandingan data training dan dataset serta menambah data pada <i>slangword</i>	Penelitian membahas tentang sentimen <i>cyberbullying</i> , menggunakan tiga rasio data splitting
10	Tren Marketplace Berdasarkan Klasifikasi Ulasan Pelanggan Menggunakan Perbandingan Kernel <i>Support Vector Machine</i>	Dwi Latifah Rianti, dkk., STRING (Satuan Tulisan Riset dan Inovasi Teknologi), 2021	Membandingkan kernel SVM pada klasifikasi ulasan pelanggan marketplace	Kernel yang memiliki akurasi terbaik adalah sigmoid. Pada penelitian ini disimpulkan bahwa pemilihan kernel SVM berpengaruh pada hasil akhir klasifikasi	Hanya menggunakan satu skenario pengujian	Kasus yang diteliti adalah <i>cyberbullying</i> pada tweet berbahasa Indonesia, menggunakan tiga skenario pengujian

2.3. Landasan Teori

2.3.1 Machine Learning

Machine learning dapat di definisikan sebagai paradigma komputasi untuk memecahkan permasalahan dengan melihat contoh-contoh kejadian sebelumnya (*training*). Dasar ide penemuan machine learning yaitu proses menalar masalah dengan mengacu pada kasus sebelumnya yang serupa. *Machine learning* dapat diterapkan untuk tugas klasifikasi, regresi, *clustering*, dan *hybrid tasks* (Taeho Jo, 2021). Aplikasi dari metode machine learning dengan database yang berukuran besar disebut data mining (Ethem Alpaydin, 2014).

Machine learning memiliki keterbatasan karena terdapat beberapa prasyarat yang harus dipenuhi dalam menyelesaikan permasalahan. Hal ini dikarenakan jika prasyarat tersebut tidak terpenuhi maka berdampak pada akurasi hasil yang menurun. Berikut beberapa kelemahan (Purba Danu Kusuma, 2020):

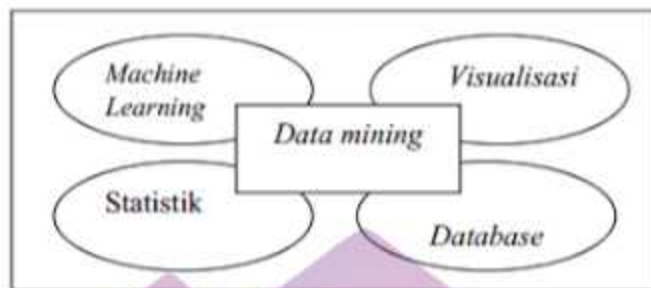
- a. Membutuhkan *database* atau memori yang besar.
- b. Membutuhkan proses pelabelan, khususnya untuk *supervised learning* yang bisa membutuhkan waktu lama.
- c. Munculnya bias sehingga hasil tidak akurat.

Metode *machine learning* terdiri dari dua pendekatan yaitu *supervised learning* dan *unsupervised learning*. Pembelajaran terbimbing adalah proses pengumpulan informasi dari serangkaian pengamatan yang hasil yang diharapkan diketahui sebelumnya. Biasanya, pembelajaran yang diawasi menggunakan data yang ada. Untuk pembelajaran terawasi, *supervisor* adalah variabel target atau kolom data yang mewakili nilai prediksi bersama dengan kolom data lainnya.

variabel target yang dipilih mewakili jawaban atas pertanyaan organisasi dan membantu membuat keputusan. Secara umum, *supervised learning* disebut juga dengan model prediktif atau model prediksi. Algoritma pemodelan prediktif utama adalah klasifikasi untuk variabel target kategoris atau regresi untuk variabel target kontinu (Abbott, 2014). Sedangkan *unsupervised learning* merupakan pembelajaran tanpa pengawasan sering disebut dengan model deskriptif atau model deskriptif tidak memiliki variabel sasaran. Data masukan dianalisis dan dikelompokkan menurut kedekatan dengan data masukan. Setiap kelompok ditandai untuk menunjukkan siapa yang termasuk dalam kelompok tersebut (Ningrum, 2018).

2.3.2 Data Mining

Data Mining merupakan ekstraksi *nontrivial* (pernyataan yang tidak sederhana atau pernyataan yang tidak mudah dibuktikan) dari informasi implisit, yang sebelumnya tidak diketahui, dan berpotensi berguna dari kumpulan data yang berukuran besar. Proses data mining biasanya memiliki urutan yaitu manajemen data, pra-pemrosesan data (*preprocessing*), *mining*, dan pengolahan setelah proses (*postprocessing*) (Tao Li, 2011). Tujuan *data mining* adalah untuk menemukan pola yang tidak diketahui sebelumnya melalui analisis matematika. *Data mining* merupakan irisan pola dari berbagai disiplin ilmu, seperti pada gambar 1 berikut (Tan, 2012):



Gambar 2. 1. Irisan Pola *Data Mining*

Istilah *data mining* dan *Knowledge Discovery Data (KDD)* sama-sama menjelaskan tentang proses penggalian informasi yang belum diketahui dalam suatu kumpulan data yang berukuran besar. Namun, konsep dari kedua proses tersebut berbeda, akan tetapi saling berkaitan. *Data mining* merupakan salah satu tahapan dalam proses KDD. KDD merupakan langkah-langkah yang bertujuan untuk menggali dan menganalisis kumpulan data dan mengekstrak informasi. Tahapan KDD adalah sebagai berikut (Fitri Marisa, 2021):

- a. *Data cleaning*: proses memeriksa data yang tidak relevan, memperbaiki kesalahan data, dan membuang duplikasi data.
- b. *Data integration*: proses menggabungkan data dari berbagai *database*.
- c. *Data selection*: proses pemilihan data.
- d. *Data transformation*: proses transformasi data ke format tertentu.
- e. *Data mining*: proses mencari pola atau informasi dengan suatu metode algoritma.
- f. *Pattern evaluation*: proses evaluasi dari model atau pola dari hasil *data mining*.

g. *Knowledge presentation*: visualisasi atau menampilkan hasil proses yang berupa pola informasi.

2.3.3 Text Mining

Text mining adalah proses untuk memperoleh informasi yang berkualitas dari teks (Onno W.Purbo, 2019). *Text mining* termasuk salah satu bagian data mining. *Text mining* adalah proses untuk mengolah data tekstual untuk mendapat pengetahuan implisit. Terdapat tiga tahapan umum dalam text mining yaitu *text preprocessing*, *feature selection*, dan *text analytic*. *Text Preprocessing* adalah proses pembersihan data untuk keperluan analisis lebih lanjut. Tahapan *text preprocessing* biasanya adalah *case folding* (format penulisan huruf kecil), *cleaning* (menghapus karakter selain alfabet), *tokenizing* (pemotongan string input), *filtering* (membuang kata yang kurang penting), *stemming* (menentukan kata dasar). *Feature selection* adalah tahapan untuk menghapus *features* yang tidak relevan dalam suatu dataset. *Text analytic* adalah tahapan mengolah data dengan algoritma untuk analisis (Fikri Adi Nugraha, dkk, 2020).

Text mining dapat diartikan sebagai proses menambang data yang berupa teks yang bersumber dari dokumen. Tujuan *text mining* adalah untuk memperoleh informasi yang berguna dari dokumen yang telah terkumpul dengan cara menganalisis keterkaitan antar dokumen. *Text mining* memiliki peran dalam pengelompokan teks dan pengkategorisasian teks (Kurnia Fitriani, dkk, 2021).

2.3.4 Analisis Sentimen

Sentymment analysis adalah suatu proses atau tahapan untuk melakukan estimasi atau mengklasifikasi teks ke dalam berbagai bentuk sentimen (positif,

netral, negatif, sarkas, dll) (Fikri Adi Nugraha et al, 2020). Analisis sentimen merupakan suatu runtutan tugas yang tetap, bertujuan untuk mengevaluasi polaritas yang muncul dalam ulasan pengguna sosial media tertentu. Pada umumnya, polaritas dapat tergantung pada perasaan yang disampaikan pengguna melalui ulasan di sosial media yang dapat berupa positif, negatif, atau netral (Ahmed A.Abd Latif et al, 2021).

Menurut (Fikri Aldi Nugraha; dkk, 2020) analisis sentimen (*opinion mining*) memiliki tugas untuk mengelompokkan polaritas pada teks apakah pendapat yang disampaikan bersifat positif, negatif, atau netral. Dalam analisis sentimen terdapat beberapa proses yaitu:

a. *Document-level sentiment analysis*

Proses ini bertujuan untuk mencari klasifikasi suatu dokumen yang menyatakan pendapat mengenai suatu topik atau target

b. *Sentence and phrase-level sentiment analysis*

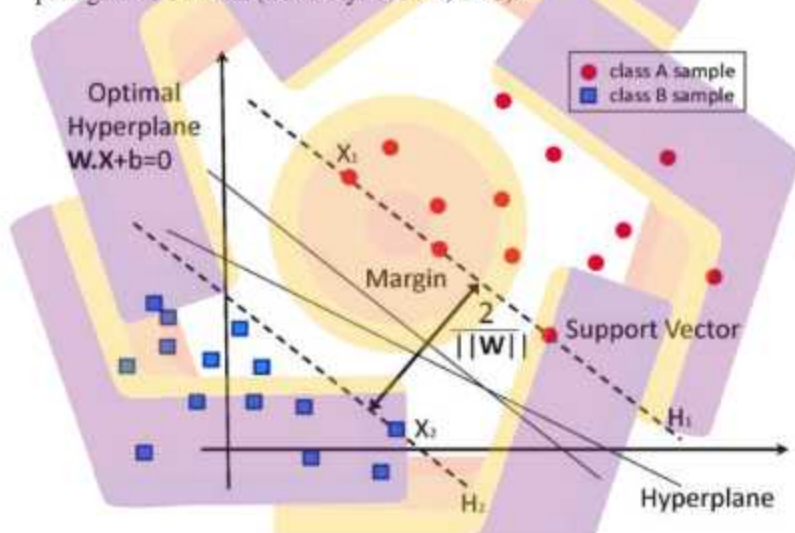
Proses ini merupakan pemecahan kalimat atau teks menjadi unit terkecil sesuai kata. Tahap ini akan memproses setiap kalimat dengan cara menganalisis satu persatu lalu diklasifikasikan ke dalam kategori sentimen yang berupa positif atau negatif.

c. *Entity and aspect-level opinions*

Tahapan ini merupakan proses mengukur sentimen dengan melihat pendapat terhadap karakteristik dari entitas.

2.3.5 Support Vector Machine (SVM)

SVM adalah salah satu metode *machine learning* yang digunakan untuk klasifikasi. Karakteristik SVM yaitu bersifat *non-probabilistic*, membagi data menjadi dua kategori atau *binary*, dan masing-masing data dibatasi oleh *hyperplane* (Rifkie Primartha, 2021). Teknik SVM digunakan untuk menemukan *hyperplane* (fungsi pemisah) yang optimal pada observasi dengan nilai variabel target yang berbeda. Hyperplane dapat berupa *line* pada data dua dimensi atau *flat plane* pada data multi dimensi (Eva Damila, dkk, 2019). Ilustrasi metode SVM dapat dilihat pada gambar 2 berikut (Dinda Ayu Pratiwi, 2020):

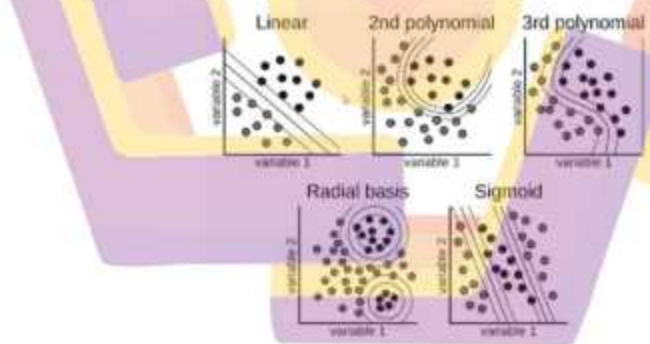


Gambar 2. 2. Support Vector Machine (SVM)

SVM memiliki beberapa keunggulan yaitu efektif digunakan untuk kasus yang memiliki dimensi tinggi, mampu melakukan klasifikasi untuk kasus yang berdimensi lebih besar dari sampel, SVM hemat memori, memiliki banyak kernel

sehingga menjadi variasi untuk digunakan pada kasus apapun (Rahmadya et al, 2020).

Namun SVM juga memiliki kelemahan yaitu terjadinya overfitting, tidak menyediakan nilai probabilitas, dan perlu mencari fungsi kernel yang tepat untuk meningkatkan *regularization* (Rahmadya et al, 2020). Ketika menghadapi masalah dengan data yang tidak dipisahkan secara linier di ruang input, SVM tepi lunak tidak dapat menemukan tingkat partisi yang kuat yang meminimalkan kesalahan klasifikasi titik data, dan menggeneralisasi dengan baik. Dengan demikian, kernel dapat digunakan untuk mengubah data menjadi ruang dimensi yang lebih tinggi yang disebut ruang kernel, mendiskritisasi data secara linier (Awad dan Khanna, 2015). Terdapat beberapa jenis kernel yang sering digunakan untuk klasifikasi, seperti pada gambar 2.3. berikut:



Gambar 2. 3. Kernel SVM

a. *Linear*

Linear kernel dapat diartikan sebagai fungsi kernel yang paling sederhana yang digunakan ketika data yang akan diolah sudah terpisah secara *linear*. Kernel ini cocok untuk banyak fungsi, karena pemetaan ke ruang dimensi yang lebih tinggi tidak meningkatkan kinerja seperti halnya untuk klasifikasi teks. Dalam klasifikasi teks, jumlah kemunculan (dokumen) dan fitur (kata) adalah sama (Kowalczyk, 2014). Adapun persamaan linear kernel sebagai berikut (Rifkie Primartha, 2021):

$$k_{x,y} = xTy + c \dots \dots \dots (1)$$

b. *Polynomial*

Polynomial kernel adalah fungsi kernel yang digunakan ketika data tidak didiskritisasi secara linier. Polinomial kernel sangat cocok untuk memecahkan masalah di mana semua data pelatihan dinormalisasi (Ningrum, 2018). Adapun persamaan *polynomial* kernel sebagai berikut (Rifkie Primartha, 2021):

$$k[x,y] = [axTy + c]d \dots \dots \dots (2)$$

c. RBF

Kernel RBF adalah fungsi kernel yang sering digunakan dalam analisis ketika data tidak dipisahkan secara linier. Kernel RBF memiliki dua parameter yaitu Gamma dan Cost. Parameter Cost, atau biasa disebut C, adalah parameter yang berfungsi sebagai pengoptimal SVM untuk menghindari kesalahan klasifikasi setiap sampel dalam set data pelatihan. Parameter gamma menentukan seberapa jauh sampel kumpulan data pelatihan dipengaruhi oleh nilai rendah yang berarti "jauh" dan nilai tinggi yang berarti "dekat". Untuk rentang kecil, titik-titik yang

jauh dari garis pemisah yang dapat diterima diperhitungkan dalam perhitungan garis pemisah. Jika rentang tinggi berarti skor berada di sekitar garis yang wajar, hitungan dianggap (Patel, 2017). Adapun persamaan *polynomial* kernel sebagai berikut (Rifkie Primartha, 2021):

$$k[x, y] = \exp[-\gamma \|x - y\|^2] \dots \dots \dots (2)$$

d. Sigmoid

Sigmoid adalah trick dari kernel SVM untuk jaringan saraf tiruan. Trick kernel memiliki beberapa kelebihan karena dalam proses pembelajaran SVM, pengguna hanya perlu mengetahui fungsi kernel trick yang digunakan untuk menentukan support vector, tanpa perlu mengetahui bentuk fungsi nonlinier. Dari semua trick kernel, trick kernel fungsi radial basis adalah trick kernel yang memberikan hasil terbaik dalam proses klasifikasi, terutama untuk data yang tidak dapat dipisahkan secara linier. Adapun persamaan sigmoid kernel sebagai berikut (Rifkie Primartha, 2021):

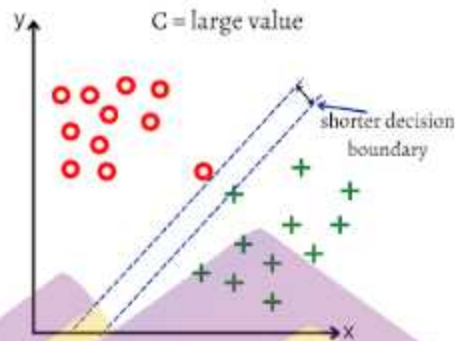
$$k[x, y] = \tanh [axTy + c] \dots \dots \dots (4)$$

Penggunaan kernel-kernel diatas juga melibatkan parameter pendukung.

Adapun beberapa parameter yang digunakan pada SVM yaitu:

a. Cost (C)

Parameter C berfungsi untuk mengontrol *trade-off* antara mendukung margin yang lebih besar dan kesalahan klasifikasi yang lebih kecil. Adapun pengaruh parameter C dapat dilihat pada gambar 2.4 berikut:



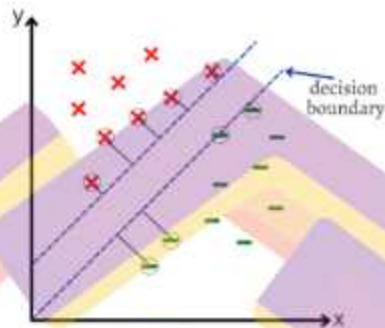
Gambar 2. 4. Parameter C

Pada gambar 2.4 dapat dilihat bahwa penggunaan nilai C memiliki pengaruh terhadap kinerja kernel. Semakin tinggi nilai C , SVM akan berusaha untuk mendapatkan margin yang lebih sempit, mengizinkan beberapa data pelatihan yang ambigu (mungkin noise) berada di dalam margin. Sebaliknya, nilai C yang lebih rendah akan mencoba untuk memaksimalkan margin, bahkan jika beberapa data pelatihan salah diklasifikasikan.

b. Gamma

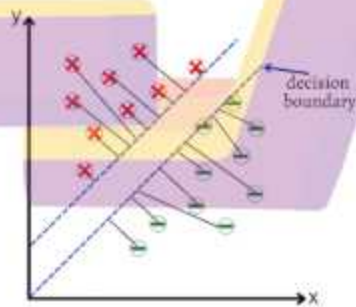
Parameter gamma merupakan salah satu parameter utama pada SVM yang menggunakan kernel. Parameter ini mengontrol bentuk dari fungsi basis radial yang digunakan untuk menghitung kedekatan antara titik data dalam ruang fitur. Memilih nilai gamma yang sesuai merupakan langkah penting dalam mengoptimasi model SVM. Nilai yang tepat akan sangat tergantung pada karakteristik data yang Anda miliki. Ada berbagai pendekatan untuk menemukan nilai gamma yang optimal, seperti melakukan penelusuran *grid* (*grid search*) atau menggunakan algoritma

optimasi seperti Random Search. Adapun ilustrasi penggunaan nilai gamma dapat dilihat pada gambar 2.5 dan gambar 2.6.



Gambar 2. 5. Parameter Gamma (Tinggi)

Pada gambar 2.5 menunjukkan penggunaan nilai gamma yang tinggi. Nilai gamma yang tinggi dapat menyebabkan model SVM menjadi lebih rumit dan cenderung overfitting, terutama jika data pelatihan tidak cukup besar.



Gambar 2. 6. Parameter Gamma (Rendah)

Pada gambar 2.6 menunjukkan penggunaan nilai gamma yang rendah. Nilai gamma yang rendah dapat menyebabkan model cenderung underfitting, karena pengaruh titik data tetangga menjadi lebih luas.

c. Degree

Parameter degree merupakan salah satu parameter yang digunakan dalam algoritma SVM. Pilihan nilai degree yang tepat akan sangat tergantung pada karakteristik data. Nilai yang terlalu rendah mungkin tidak mampu menangkap pola yang rumit, sementara nilai yang terlalu tinggi dapat menyebabkan overfitting. Pada umumnya, eksperimen dengan beberapa nilai degree yang berbeda diperlukan untuk menemukan nilai yang optimal, dan validasi silang juga dapat membantu memilih nilai terbaik.

2.3.6 TF-IDF

Term Frequency Inverse Document Frequency (TF-IDF) merupakan metode untuk memberi bobot suatu kata pada dokumen. Kegunaan TF-IDF yaitu untuk melakukan evaluasi tentang seberapa penting *term* pada dokumen (I Putu Gede Hendra Suputra, dkk, 2019). *Term Frequency* (TF) adalah jumlah kemunculan sebuah kata dalam dokumen. TF dapat dihitung dengan persamaan 5 berikut:

$$TF_{term, doc} = \frac{\text{jumlah kemunculan term dalam dokumen}}{\text{jumlah total term dalam dokumen}} \dots \dots \dots (5)$$

Inverse Document Frequency (IDF) merupakan ukuran kesamaan kata di semua dokumen. IDF dapat dihitung dengan persamaan 6 berikut:

$$IDF_{term} = \log\left(\frac{\text{jumlah total dokumen}}{\text{jumlah dokumen yang mengandung term}}\right) \dots \dots \dots (6)$$

Persamaan TF-IDF didapat dari persamaan TF dan IDF. Rumus TF-IDF dapat dilihat pada persamaan 7 berikut:

$$TF - IDF = TF \times IDF \dots \dots \dots (7)$$

Kisaran nilai TF-IDF antara 0 hingga 1. Nilai 0 berarti sebuah kata tidak memiliki kepentingan begitupun sebaliknya, nilai 1 berarti memiliki kepentingan total pada dokumen (Said R.K.Bahasyim, dkk, 2021)

2.3.7 Confusion Matrix

Confusion matrix merupakan matriks yang menunjukkan visualisasi kinerja algoritma klasifikasi menggunakan data dalam matriks. *Confusion matrix* membandingkan klasifikasi yang diprediksi dengan klasifikasi aktual (Profost, dkk, 1998). *Confusion matrix* dapat disebut sebagai tabel yang menjelaskan tentang klasifikasi jumlah data uji yang benar dan klasifikasi jumlah data uji yang salah (Dwi Normawati, dkk, 2021). Model *confusion matrix* dapat dilihat pada tabel 2 (Banu Putri Pratiwi, dkk, 2020).

Tabel 2. 2. *Confusion Matrix*

	<i>Observed</i>	
	<i>True</i>	<i>False</i>
<i>Predicted Class</i>	<i>True</i>	<i>False</i>
<i>True</i>	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
<i>False</i>	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Keterangan:

1. TP merupakan jumlah data positif yang terklasifikasi benar.
2. TN merupakan jumlah data negatif yang terklasifikasi benar.
3. FN merupakan jumlah data negatif yang terklasifikasi salah.

4. FP merupakan jumlah data positif yang terklasifikasi salah.

Akurasi merupakan perbandingan data yang terklasifikasi benar dengan total data. Akurasi dapat diperoleh dengan persamaan 4 berikut:

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \dots\dots\dots (8)$$

Presisi merupakan jumlah data positif yang terklasifikasi benar dibagi total data yang terklasifikasi positif. Presisi dapat diperoleh dengan persamaan 5 berikut:

$$\text{Presisi} = \frac{TP}{TP+FP} \times 100\% \dots\dots\dots (9)$$

Recall merupakan beberapa persen data positif yang terklasifikasi benar. Recall dapat diperoleh dengan persamaan 6 berikut:

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% \dots\dots\dots (10)$$

F1-Score (F-Measure) merupakan nilai yang didapat dari Presisi dan *Recall* antara kategori hasil prediksi dengan kategori data sesungguhnya. *F1-Score* diperoleh dari persamaan 7 berikut:

$$\text{F1-Score} = \frac{2 \times \text{Recall} \times \text{Presisi}}{\text{Recall} + \text{Presisi}} \times 100\% \dots\dots\dots (11)$$

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis penelitian yang digunakan dalam penelitian ini adalah eksperimental dan studi kasus. Penelitian eksperimen merupakan penelitian yang melakukan serangkaian proses untuk membuktikan konsep. Dalam hal ini peneliti melakukan eksperimen terhadap data *tweet* yang dikumpulkan melalui proses *crawling* untuk selanjutnya diklasifikasi menggunakan metode SVM. Penelitian ini berjenis studi kasus karena dilakukan di media sosial twitter.

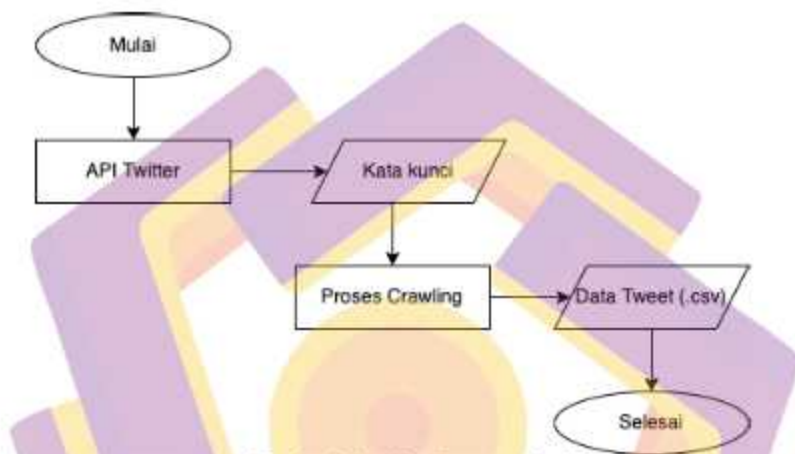
Penelitian ini bersifat deskriptif, dimana peneliti menggunakan data yang terstruktur atau dapat dijelaskan menggunakan tabel/diagram dan angka. Data *tweet* yang telah terkumpul disajikan dalam tabel yang terdiri dari kolom *tweet* dan kolom kategori. Hasil dari penelitian ini berupa nilai akurasi dari penerapan SVM sehingga disajikan dalam bentuk angka melalui pengujian *confusion matrix*.

Penelitian ini menerapkan pendekatan kuantitatif karena dalam proses penelitian menggunakan perhitungan matematis untuk mencapai hasil yang diinginkan. Perhitungan yang dimaksud adalah pada proses pengujian menggunakan *confusion matrix*.

3.2. Metode Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data primer yang didapat dari *crawling twitter*. Proses *crawling* dilakukan dengan pemrograman Python dengan menggunakan kata kunci penyebab munculnya cyberbullying. Kata

kunci tersebut telah ditentukan dari hasil wawancara dengan ahli Bahasa yang dilakukan oleh peneliti sebelumnya (Christevan Destisus, dkk, 2020). Data hasil *crawling* akan diambil 50 tweet untuk masing-masing kata kunci. Adapun alur pengumpulan data dapat dilihat pada gambar 3.1 berikut:



Gambar 3. 1. Alur Pengumpulan Data

3.3. Metode Analisis Data

Dalam penelitian ini, peneliti menggunakan Microsoft Excel dan Python. Peneliti menggunakan metode analisis data sebagai berikut:

- a. Analisis deskriptif untuk melakukan analisis awal atau untuk mencari gambaran umum terhadap keberadaan *tweet* yang mengandung *cyberbullying* atau tidak
- b. Analisis sentimen dengan metode SVM untuk melakukan klasifikasi *tweet* yang mengandung unsur *cyberbullying* dan *non-cyberbullying*.

3.4. Alur Penelitian

Dalam menjawab rumusan masalah yang telah ditentukan, maka langkah pertama yang akan dilakukan adalah mengumpulkan data *tweet* melalui proses *crawling*. Setelah data berhasil terkumpul maka akan menjadi suatu basis data sehingga dapat dilanjutkan ke langkah selanjutnya yaitu tahap *preprocessing*. Adapun tahapan *preprocessing* yang akan dilakukan penulis sebagai berikut:

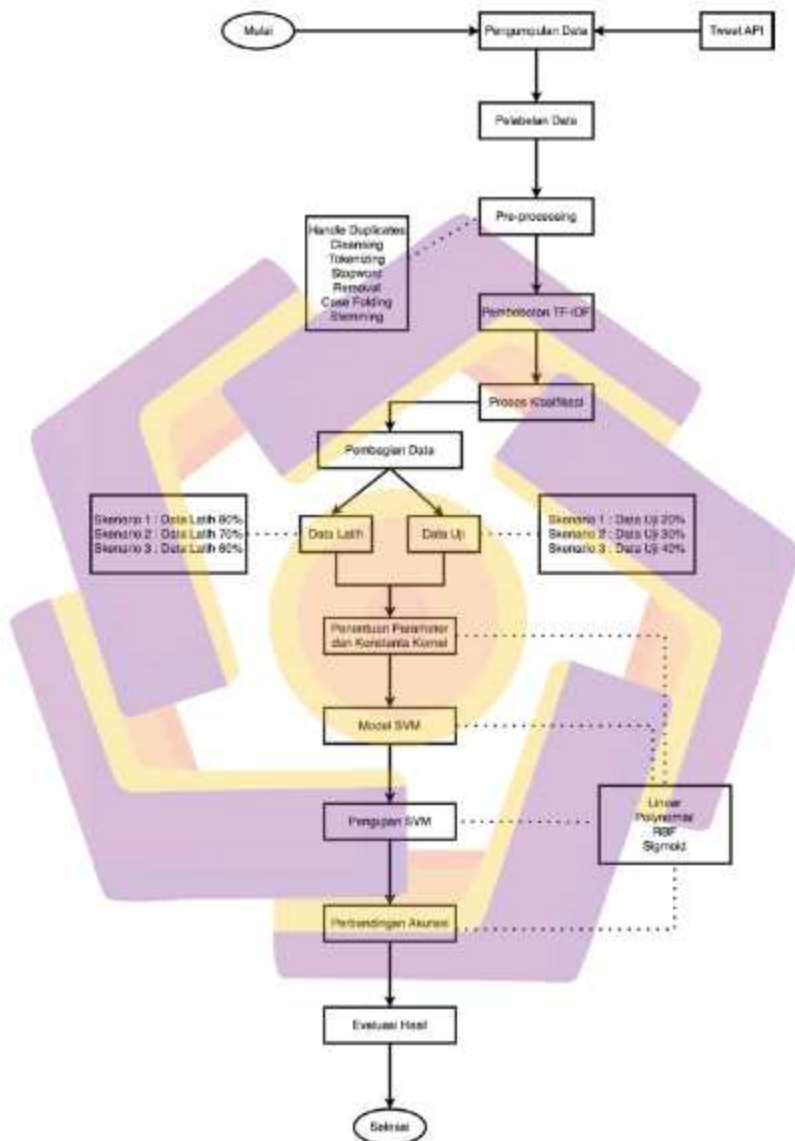
- a. *Handle duplicates* merupakan tahap yang akan dilakukan penulis untuk menghapus baris data ganda atau duplikat pada *dataset*.
- b. Pelabelan data merupakan tahap yang akan dilakukan penulis untuk memberikan label pada data *tweet* menjadi dua kategori yaitu *cyberbullying* dan *non-cyberbullying*.
- c. *Cleansing* merupakan tahap yang akan dilakukan penulis untuk merapikan data dengan memperbarui atau membenarkan data.
- d. *Tokenizing* merupakan tahap yang akan dilakukan penulis untuk memisahkan kalimat menjadi token atau potongan kata.
- e. *Stopword Removal* merupakan tahap yang akan dilakukan penulis untuk mengambil kata-kata penting atau yang akan digunakan.
- f. *Case folding* merupakan tahap yang akan dilakukan penulis untuk menyamakan karakter yaitu mengubah huruf menjadi huruf kecil.
- g. *Stemming* merupakan tahap yang akan dilakukan penulis untuk membentuk kata dasar. Pada tahap ini penulis menggunakan *library* Sastrawi.

Setelah melakukan tahap *preprocessing*, penulis akan melakukan pembobotan TF-IDF. Tujuan TF-IDF adalah untuk mengubah kata menjadi bentuk

angka. Selanjutnya penulis akan melakukan klasifikasi menggunakan metode SVM dengan menerapkan empat kernel. Kernel yang digunakan yaitu linear, polynomial, RBF, dan sigmoid.

Pada pengimplementasian metode SVM akan dilakukan *training* dan *testing* terhadap data. Skenario pertama yaitu data hasil dari *preprocessing* akan dibagi menjadi 2 kategori yaitu data *training* dan data *testing*. Data training akan menggunakan 80% data, sedangkan 20% data lainnya akan digunakan sebagai data *testing*. Eksperimen kedua dilakukan dengan mengubah skenario pembagian data. Data hasil dari *preprocessing* akan tetap dibagi menjadi 2 kategori yaitu data *training* dan data *testing*, namun jumlah data yang digunakan berbeda porsi. Data yang digunakan untuk *training* adalah 70% sedangkan data *testing* adalah 30%. Eksperimen ketiga juga dilakukan dengan mengubah skenario pembagian data. Pengkategorian data tetap sama seperti dua skenario sebelumnya. Data yang digunakan untuk *training* adalah 60% sedangkan data *testing* adalah 40%.

Hasil dari klasifikasi akan dievaluasi melalui pengujian menggunakan *confusion matrix*. Setelah seluruh skenario dilakukan, langkah terakhir adalah melakukan komparasi atau membandingkan hasil untuk mencari kernel yang mempunyai nilai akurasi tertinggi sehingga bisa ditarik kesimpulan. Skenario alur penelitian dapat dilihat pada gambar 3.2 berikut.



Gambar 3. 2. Alur Penelitian

sehingga sulit untuk langsung diolah. Oleh karena itu, pada penelitian ini perlu dilakukan seleksi supaya data menjadi lebih terstruktur atau memiliki keseragaman untuk dianalisis pada proses mining. Peneliti menggunakan baris kode dengan bahasa Python untuk melakukan proses *pre-processing*.

4.2.1 Handle Duplicates

Handle Duplicates dilakukan untuk mengatasi data ganda pada sekumpulan data. Pengecekan duplikasi data dilakukan dengan memanfaatkan bahasa pemrograman Python seperti pada gambar 4.3 berikut.

```
df = dt.drop_duplicates(subset=['data'], keep='last', inplace=False)
df.data = df.data.astype(str)
df
```

Gambar 4. 3. Handle Duplicates

Script yang terdapat pada gambar 4.3 di atas merupakan langkah-langkah untuk menghapus data yang ada pada kolom dataset yang bernama data dalam *data frame* yang bernama dt. Setelah itu, dataset akan diubah kedalam bentuk tipe data string lalu akan disimpan ke data frame baru yang bernama dt. Untuk menjalankan script pada gambar 4.3 peneliti melakukan import modul pandas.

4.2.2 Pelabelan Data

Sekumpulan data yang akan digunakan untuk analisis diberi label secara manual. Data tersebut dibagi menjadi dua kategori yaitu data *cyberbullying* dan *non-cyberbullying*. Adapun contoh pelabelan data seperti pada tabel 4.1 berikut.

Tabel 4. 1. Contoh Pelabelan Dataset

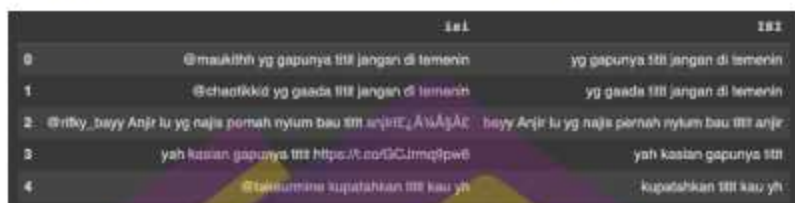
Tweet	Kategori
Woy njing!! ibumu pelacur ya pantasen kelakuan lu bejad! Dasar anak haram!! MekkelSunge bruakakakak :D	Cyberbullying
@six99one @ciloqcilq miakin literasi lu	Cyberbullying
"Kenapa kelahiran anak karena Zina disebut anak haram? Bukankah yang haram adalah perbuatan orang tuanya?"	Non-cyberbullying
gua siyap jelek nilainya	Non-cyberbullying

Pelabelan data pada tabel 4.1 dilakukan oleh psikolog dengan memperhatikan unsur konteks dan niat dalam suatu kalimat tweet. Konteks dan niat yang dimaksud yaitu mengidentifikasi arti kalimat tweet apakah mengandung motif untuk merundung seseorang atau tidak. Jika kalimat tweet berbahasa kasar atau mencela tetapi tidak bertujuan untuk merundung atau mengganggu psikologi pembaca atau target korban maka kalimat tersebut dapat dikategorikan sebagai data *non-cyberbullying*. Hal ini dikarenakan kalimat celaan tersebut bisa jadi merupakan suatu bentuk kritik atau keluh kesah terhadap diri dari penulis tweet. Sedangkan jika kalimat tweet mengandung motif tindakan perundungan atau ditujukan untuk target korban yang secara psikologi mengganggu target korban maka kalimat tersebut dikategorikan sebagai data *cyberbullying*.

4.2.3 Cleansing

Cleansing dalam penelitian ini digunakan untuk menghilangkan karakter yang tidak dibutuhkan pada saat proses mining. Proses cleansing dilakukan dengan

memuliskan karakter apa saja yang akan dihilangkan pada *source code*. Pada penelitian ini peneliti membersihkan karakter tanda baca dan menghapus teks URL. Contoh hasil *cleansing* dapat dilihat pada gambar 4.4 berikut.



Gambar 4. 4. Hasil Cleansing

4.2.4 Tokenizing

Tahap tokenizing dilakukan untuk memotong dokumen menjadi bagian-bagian kata atau token. Proses pembentukan toke menggunakan *Natural Language Toolkit* (NLTK) package yang berbasis bahasa Indonesia. Sedangkan hasil tokenizing dapat dilihat pada gambar 4.5 berikut.



Gambar 4. 5. Tokenizing

4.2.5 Stopword Removal

Pada tahap *stopword removal*, kata-kata yang dianggap penting akan dipilih. Proses ini memanfaatkan algoritma *stoplist* untuk membuang kata-kata yang kurang penting. Selain itu proses ini juga memanfaatkan algoritma *wordlist* untuk

menyimpan kata-kata penting atau kata yang akan digunakan untuk *mining*. Script untuk stopwords removal dapat dilihat pada gambar 4.6 berikut.

```
#stopremoval

stopword = nltk.corpus.stopwords.words('indonesian')

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

df['STOP_REMOVAL'] = df['TOKENIZATION'].apply(lambda x: remove_stopwords(x))
df.head(5)
```

Gambar 4. 6. Script Stopword Removal

4.2.6 Case Folding

Case Folding digunakan pada penelitian ini untuk menyeragamkan karakter pada dataset. Karakter yang diseragamkan pada kasus ini yaitu menyamakan huruf menjadi huruf kecil.

4.2.7 Stemming

Tahap *stemming* digunakan untuk membentuk kata dasar pada setiap kata pada dataset. Contoh dalam proses ini terdapat kata "berseragam" maka akan diubah menjadi "seragam". *Library* yang digunakan untuk *stemming* yaitu Sastrawi.

4.3 Klasifikasi Support Vector Machine

4.3.1 Pembagian Data Latih dan Data Uji

Pembentukan data latih (*data train*) dibutuhkan untuk melatih kinerja dalam pembelajaran mesin (*machine learning*). Pada tahap *training* sebagian data yang

telah diketahui kategorinya digunakan untuk pemodelan. Pembuatan data latih dilakukan dengan mengklasifikasi seluruh data dengan cara manual. Data tersebut dikategorikan menjadi *cyberbullying* atau *non-cyberbullying*. Menurut saran (Dwi Latifah Rianti, dkk, 2021) saat melakukan pengujian sebaiknya menggunakan beberapa skenario pembagian data latih dan data uji. Sehingga pada penelitian ini, peneliti akan menggunakan tiga skenario pembagian data atau pengujian. Pembagian data latih dan data uji dapat dilihat pada tabel 4.2 berikut.

Tabel 4. 2. Pembagian Data

Opsi	Jumlah Dataset	Presentase Data Latih	Presentase Data Uji
Skenario 1	1850	60%	40%
Skenario 2		70%	30%
Skenario 3		80%	20%

4.3.2 Eksperimen dan Evaluasi Support Vector Machine

1. Ruang Lingkup Eksperimen

Pada penelitian ini peneliti menggunakan perangkat keras dan perangkat lunak untuk eksperimen analisis *cyberbullying*. Ruang lingkup eksperimen ini berupa komputer yang memiliki spesifikasi seperti pada tabel 4.3 berikut.

Tabel 4. 3. Perangkat Eksperimen

Perangkat	Spesifikasi
Perangkat Keras	Processor 1,8 GHz Dual-Core Intel Core i5 Memory 8GB
Perangkat Lunak	Sistem Operasi: macOS Catalina versi 10.15.6 Perangkat lunak pengembang: Google Colaboratory, Google Drive, Google Sheet Perangkat Pembantu: Microsoft Excell

2. Ekstraksi Fitur TF-IDF

Peneliti melakukan tahap ekstraksi fitur menggunakan TF-IDF. Tahap ini bertujuan untuk menghitung bobot relatif dari sebuah kata dalam dataset hasil *preprocessing*. Selain itu TF-IDF dapat digunakan dalam pemrosesan teks dan pengambilan informasi untuk memberikan penekanan pada kata-kata yang dianggap penting dalam sebuah dokumen. Peneliti mengimplementasikan TF-IDF dengan bantuan bahasa pemrograman python. Adapun *script* yang digunakan dapat dilihat pada gambar 4.7 berikut.

```
Tfidf_vect = TfidfVectorizer()
Tfidf_vect.fit(df['data'])

Train_X_Tfidf = Tfidf_vect.transform(Train_X)
Test_X_Tfidf = Tfidf_vect.transform(Test_X)

Train_X_Tfidf.shape

Train_Y.shape

print('TF-IDF ', type(Test_X_Tfidf), Train_X_Tfidf.shape)

SVM = SVC()
cross_val_score(SVM,Train_X_Tfidf,Train_Y, cv=100)
```

Gambar 4. 7. Skenario 1 TF-IDF

Script pada gambar 4.7 menjelaskan bahwa untuk menerapkan TF-IDF peneliti membuat objek `TfidfVectorizer` yang digunakan untuk menghitung bobot dari teks. Data yang akan dieksekusi yaitu dataset pada kolom data dalam data *frame* yang diberi nama `df`. Setelah peneliti melakukan *fitting* pada `TfidfVectorizer`, peneliti menggunakan metode `transform` untuk mengubah teks dalam data pelatihan (`Train_X`) menjadi representasi TF-IDF. Hal ini menghasilkan matriks dengan dimensi (jumlah dokumen pelatihan, jumlah fitur), di mana setiap entri dalam

matriks mewakili bobot TF-IDF untuk kata pada posisi yang sesuai dalam vektor fitur. Peneliti menggunakan metode transform yang sama untuk mengubah teks dalam data uji (Test_X) menjadi representasi TF-IDF yang sesuai. Ini memastikan bahwa representasi fitur untuk data uji konsisten dengan yang telah dihitung pada data pelatihan. Ekstraksi fitur TF-IDF ini diuji menggunakan tiga skenario yaitu pembagian data latih sebanyak 80%, 70%, dan 60%. Hasil TF-IDF dapat dilihat pada gambar 4.8, gambar 4.9, dan gambar 4.10.

```
SVM = SVC()
cross_val_score(SVM,Train_X_Tfidf,Train_Y, cv=100)

array([0.8      , 0.86666667, 1.      , 0.93333333, 0.86666667,
       0.73333333, 0.86666667, 1.      , 0.86666667, 0.73333333,
       0.86666667, 0.8      , 0.86666667, 0.93333333, 0.86666667,
       0.8      , 0.73333333, 0.66666667, 0.93333333, 0.86666667,
       0.86666667, 1.      , 0.8      , 0.86666667, 0.86666667, 0.86666667,
       0.93333333, 0.46666667, 1.      , 0.86666667, 0.93333333,
       0.8      , 0.8      , 0.86666667, 0.86666667, 0.86666667,
       0.86666667, 0.86666667, 1.      , 0.86666667, 0.93333333,
       1.      , 0.8      , 1.      , 0.8      , 0.93333333,
       0.73333333, 0.8      , 1.      , 1.      , 0.93333333,
       0.86666667, 0.86666667, 0.86666667, 0.86666667, 0.8      ,
       0.86666667, 0.73333333, 0.8      , 0.93333333, 0.8      ,
       0.93333333, 0.93333333, 1.      , 1.      , 0.86666667,
       0.73333333, 1.      , 0.8      , 0.86666667, 0.8      ,
       0.93333333, 0.92233333, 0.8      , 0.86666667, 0.93333333,
       0.93333333, 0.8      , 0.86666667, 1.      ,
       1.      , 0.71428571, 0.78571429, 0.92857143, 1.      ,
       0.92857143, 0.85714286, 0.92857143, 0.92857143, 1.      ,
       0.78571429, 1.      , 0.78571429, 0.85714286, 0.92857143,
       1.      , 0.78571429, 0.85714286, 0.92857143, 1.      ]])

cross_val_score(SVM,Train_X_Tfidf,Train_Y, cv=100).mean()
0.8750000000000002
```

Gambar 4. 8. Hasil TF-IDF Skenario 80:20

```

SVM = SVC()
cross_val_score(SVM,Train_X_Tfidf,Train_Y, cv=100)

array([0.84615385, 0.89230769, 0.78923077, 0.92307692, 0.84615385,
0.84615385, 0.92307692, 0.78923077, 0.76923077, 0.76923077,
0.84615385, 0.92307692, 0.92307692, 0.89230769, 0.92307692,
0.84615385, 0.92307692, 0.92307692, 0.76923077, 0.76923077,
1., 0.84615385, 0.78923077, 0.84615385, 0.84615385,
0.84615385, 0.92307692, 0.92307692, 0.89230769, 0.81538462, 1.,
0.76923077, 0.92307692, 0.84615385, 0.84615385, 0.84615385,
0.84615385, 0.84615385, 0.92307692, 0.92307692, 0.92307692,
0.92307692, 0.84615385, 0.92307692, 0.92307692, 0.89230769,
0.92307692, 0.84615385, 1., 0.53846154, 0.92307692,
0.76923077, 0.69230769, 1., 0.92307692, 0.92307692,
0.84615385, 0.92307692, 0.92307692, 0.92307692,
0.84615385, 0.78923077, 0.92307692, 0.84615385, 0.84615385,
0.84615385, 0.84615385, 0.92307692, 0.92307692, 0.92307692,
0.92307692, 0.69230769, 0.69230769, 0.69230769, 1., 1.,
0.78923077, 0.84615385, 1., 0.89230769, 0.84615385,
0.76923077, 0.84615385, 0.92307692, 0.92307692, 0.84615385,
0.92307692, 0.84615385, 1., 0.92307692, 0.84615385,
0.84615385, 0.78923077, 0.84615385, 0.84615385, 0.92307692,
0.50333333, 0.83333333, 0.91666667, 1., 1.])

cross_val_score(SVM,Train_X_Tfidf,Train_Y, cv=100).mean()

0.958713849178487

```

Gambar 4. 9. Hasil TF-IDF Skenario 70:30

```

SVM = SVC()
cross_val_score(SVM,Train_X_Tfidf,Train_Y, cv=100)

array([0.91666667, 0.83333333, 0.75, 0.91666667,
0.91666667, 0.91666667, 0.92555556, 0.83333333, 0.92555556,
0.90909091, 0.90909091, 0.81818182, 0.72727273, 0.90909091,
0.72727273, 1., 0.72727273, 0.90909091, 0.63636364,
0.72727273, 0.90909091, 0.72727273, 0.72727273, 1.,
0.81818182, 0.90909091, 0.90909091, 0.72727273, 0.81818182,
0.90909091, 0.81818182, 0.81818182, 0.72727273, 0.63636364,
0.90909091, 0.90909091, 0.72727273, 0.90909091, 0.81818182,
1., 0.81818182, 0.81818182, 0.81818182, 0.90909091,
0.90909091, 0.72727273, 0.72727273, 0.81818182, 0.72727273,
0.72727273, 0.90909091, 0.90909091, 0.90909091, 0.81818182,
0.90909091, 0.90909091, 1., 1., 0.72727273,
0.72727273, 0.72727273, 0.81818182, 0.90909091, 0.81818182,
0.81818182, 0.43636364, 1., 0.72727273, 0.81818182,
0.90909091, 0.81818182, 0.81818182, 0.83090909, 0.72727273,
0.81818182, 1., 0.90909091, 0.81818182, 0.54545455,
1., 0.90909091, 1., 1., 0.90909091,
1., 0.90909091, 0.90909091, 0.90909091, 0.81818182, 0.72727273,
0.72727273, 0.72727273, 1., 0.81818182, 0.90909091,
0.54545455, 0.90909091, 0.90909091, 1., 1., 1.])

cross_val_score(SVM,Train_X_Tfidf,Train_Y, cv=100).mean()

0.8464393939393937

```

Gambar 4. 10. Hasil TF-IDF Skenario 60:40

Pada gambar 4.8, gambar 4.9, dan gambar 4.10 dapat diketahui bahwa hasil skor TF-IDF pada setiap skenario memiliki hasil yang berbeda. Skenario 1

menggunakan data latih 80% dan data uji 20% menghasilkan skor rata-rata 87,60%. Skenario 2 menggunakan data latih 70% dan data uji 30% menghasilkan skor rata-rata 85,87%. Skenario 3 menggunakan data latih 60% dan data uji 40% menghasilkan skor rata-rata 84,64%.

3. Penentuan Parameter SVM

Evaluasi dilakukan dengan eksperimen fungsional terhadap tingkat akurasi, presisi, recall, dan f1-score. Eksperimen pada penelitian ini menggunakan tiga skenario pelatihan dan pengujian. Penerapan SVM untuk klasifikasi *cyberbullying* dilakukan dengan kernel linear, polynomial, RBF dan sigmoid. Uji coba tersebut dilakukan dengan memperhatikan parameter. Setiap kernel yang digunakan memiliki perbedaan parameter. Adapun parameter yang digunakan sebagai berikut:

- a. Kernel linear tidak memiliki parameter khusus sehingga menggunakan parameter umum SVM yaitu C (nilai margin error), tol (toleransi untuk konvergensi), dan max_iter (maksimum iterasi).
- b. Kernel polynomial melibatkan parameter $degree$, $coef0$, dan $gamma$ dalam melakukan klasifikasi.
- c. Kernel RBF memiliki parameter khusus yaitu $gamma$.
- d. Kernel sigmoid memiliki parameter khusus yaitu $gamma$ dan $coef0$.

Penentuan nilai parameter *sequential training* SVM dilakukan dengan menguji beberapa nilai untuk setiap parameter. Nilai yang digunakan sebagai nilai parameter untuk klasifikasi merupakan nilai yang memiliki akurasi tertinggi.

Setelah semua parameter mendapatkan nilai, nilai tersebut digunakan untuk setiap kernel sesuai dengan kebutuhan di setiap kernel dengan besaran nilai yang sama.

Adapun nilai parameter yang digunakan sebagai berikut:

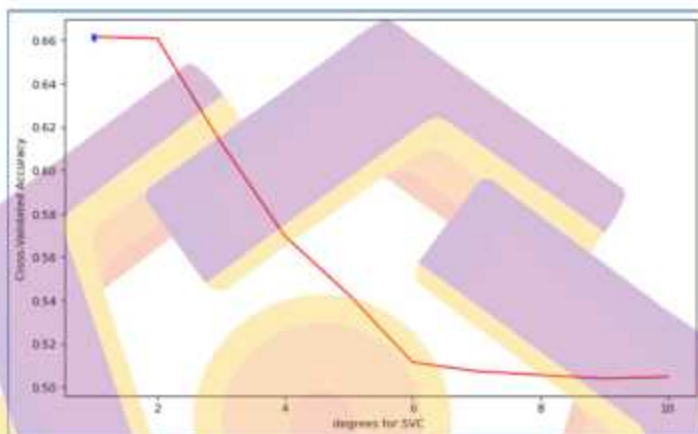
- a. Penentuan nilai degree dilakukan melalui pengujian degree dengan range satu sampai sepuluh. Tahap pertama dalam pengujian degree dilakukan dengan menginisialisasi daftar nilai degree yaitu [1,2,3,4,5,6,7,8,9,10]. Selanjutnya peneliti melakukan iterasi pada setiap degree dengan menggunakan perulangan. Pada tahap iterasi, peneliti menerapkan kernel polinomial karena kernel ini yang memiliki parameter khusus untuk menggunakan degree. Kemudian hasil iterasi akan dilakukan cross-validation dengan mengimplementasikan validasi silang dengan 10 lipatan atau $cv=10$ dan menghitung skor akurasi setiap degree. Hasil akurasi yang diperoleh disimpan dalam list yang diberi nama variabel `acc_score`. Script yang digunakan peneliti dapat dilihat pada gambar 4.11 berikut:

```
degree=[1,2,3,4,5,6,7,8,9,10]
acc_score=[]
for d in degree:
    svm = SVC(kernel='poly', degree=d)
    scores = cross_val_score(svm, test_tf, dfnew, 'silai', cv=10, scoring='accuracy')
    acc_score.append(scores.mean())

figure(num=None, figsize=(10, 6), dpi=80, facecolor='w', edgecolor='k')
plt.plot(degree, acc_score, color='r')
plt.xlabel('degrees for SVC ')
plt.ylabel('Cross-Validated Accuracy')
```

Gambar 4. 11. Pengujian Nilai Degree

Dari pengujian nilai degree yang telah dilakukan seperti pada gambar 4.11 didapatkan hasil yang disajikan dalam bentuk grafik. Dalam pembuatan grafik ini peneliti memanfaatkan *library* matplotlib. Adapun hasil pengujian dapat dilihat pada gambar 4.12 berikut.



Gambar 4. 12. Hasil Pengujian Nilai degree

Pada gambar 4.12 di atas dapat diketahui bahwa penentuan nilai degree berpengaruh terhadap *cross validate accuracy*. Masing-masing *cross validate accuracy* dari degree dapat dilihat pada tabel 4.4 berikut:

Tabel 4. 4. Penentuan Degree

No	Nilai Degree	Cross Validate Accuracy
1	1	0,662
2	2	0,66
3	3	0,63
4	4	0,57
5	5	0,54
6	6	0,51
7	7	0,51
8	8	0,51
9	9	0,51
10	10	0,51

Dari tabel 4.4 peneliti memilih degree yang memiliki nilai akurasi tertinggi. Sehingga nilai parameter yang diterapkan untuk parameter degree adalah 1. Degree 1 mendapat nilai *cross validate accuracy* 0,66%.

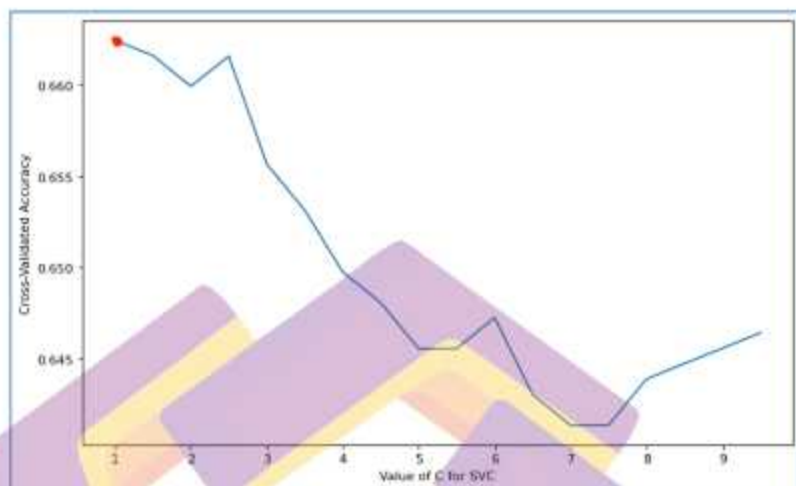
- b. Penentuan nilai *complexity* (C) dilakukan dengan melakukan perulangan dengan menggunakan beberapa nilai C pada model SVM. Kemudian, peneliti menghitung akurasi *cross-validate* menggunakan fungsi `cross_val_score()` dan menyimpan nilai akurasi rata-rata pada variabel `acc_score`. Selanjutnya, peneliti membuat plot untuk melihat bagaimana pengaruh nilai C terhadap akurasi *cross validated*. Script yang digunakan untuk menguji nilai C dapat dilihat pada gambar 4.13 berikut.

```
C_range=list(np.arange(1,10,0.5))
acc_score=[]
for c in C_range:
    svc = SVC(kernel='linear', C=c)
    scores = cross_val_score(svc, text_tf, dfnew['nilai'], cv=10, scoring='accuracy')
    acc_score.append(scores.mean())

C_values=list(np.arange(1,10,0.5))
figure=plt.figure(figsize=(10, 6), dpi=80, facecolor='w', edgecolor='k')
plt.plot(C_values,acc_score)
plt.xticks(np.arange(1,10,1))
plt.xlabel('Value of C for SVC')
plt.ylabel('Cross-Validated Accuracy')
```

Gambar 4. 13. Pengujian Nilai C

Script 4.13 dapat berjalan ketika sudah melakukan *import* modul `numpy`, `matplotlib.pyplot`, `sklearn.svm.SVC`, dan `sklearn.model_selection.cross_val_score`. Selain itu peneliti menyediakan data yang akan digunakan yaitu `text_tf` sebagai fitur dan `dfnew['nilai']` sebagai target untuk melakukan *cross-validation*. Adapun hasil *cross validate accuracy* dapat dilihat pada gambar 4.14 berikut.



Gambar 4. 14. Hasil Pengujian Nilai C

Pada gambar 4.14 dapat dilihat bahwa terdapat nilai interval 1 sampai 10 yang ada pada sumbu x, sedangkan sumbu y menyajikan nilai akurasi yang didapat. Dari gambar tersebut dapat diketahui bahwa nilai C yang memiliki akurasi terbaik adalah C dengan nilai 1. Adapun rincian nilai hasil pengujian dapat dilihat pada tabel 4.5 berikut.

Tabel 4. 5 Penentuan C

No	Nilai C	Cross Validate Accuracy
1	1	0,663
2	2	0,660
3	3	0,650
4	4	0,649
5	5	0,646
6	6	0,648
7	7	0,642
8	8	0,645
9	9	0,646
10	10	0,648

- c. Penentuan nilai *gamma* dilakukan dengan menguji range nilai gamma 0,1 hingga nilai 5. Pengujian dilakukan dengan menginisialisasi *gamma_range* sebagai daftar nilai gamma yang dihasilkan menggunakan fungsi *np.arange()*. Kemudian, peneliti membuat tempat menyimpan skor akurasi rata_rata dari setiap pengujian model dengan cara membuat list kosong yang diberi nama variabel *acc_score*. Setelah itu, melakukan iterasi untuk setiap *gamma_range* dengan menggunakan perulangan. Pada setiap iterasi, peneliti membuat objek SVC (*Support Vector Classifier*) dengan menerapkan tiga kernel yang menggunakan parameter khusus gamma untuk pemodelan, yaitu kernel RBF, kernel sigmoid dan kernel polynomial. Langkah selanjutnya yaitu melakukan cross-validation dengan menggunakan fungsi *cross_val_score()*. Fungsi ini akan melakukan validasi silang dengan 10 lipatan atau yang biasa disebut *cv=10* dan menghitung skor akurasi yang didapat pada setiap lipatan. *Source code* untuk pengujian nilai gamma dapat dilihat pada gambar 4.15 dan gambar 4.16, dan gambar 4.17 berikut.

```

gamma_range=list(np.arange(0.1,5,0.1))
acc_score=[]
for g in gamma_range:
    svc = SVC(kernel='rbf', gamma=g)
    scores = cross_val_score(svc, text_tf, dfnew['nilai'], cv=10, scoring='accuracy')
    acc_score.append(scores.mean())

figure(num=None, figsize=(10, 6), dpi=80, facecolor='w', edgecolor='k')
plt.plot(gamma_range,acc_score)
plt.xlabel('Value of gamma for SVC ')
plt.xticks(np.arange(0.1,5,0.5))
plt.ylabel('Cross-Validated Accuracy')

```

Gambar 4. 15. Pengujian Nilai Gamma RBF

```

C_range=list(np.arange(1,10,0.5))
acc_score=[]
for c in C_range:
    svc = SVC(kernel='sigmoid', C=c)
    scores = cross_val_score(svc, test_tf, dfnew['nilai'], cv=10, scoring='accuracy')
    acc_score.append(scores.mean())

C_values=list(np.arange(1,10,0.5))
figure(num=None, figsize=(10, 6), dpi=80, facecolor='w', edgecolor='k')
plt.plot(C_values, acc_score)
plt.xticks(np.arange(1,10,1))
plt.xlabel('Value of C for SVC')
plt.ylabel('Cross-Validated Accuracy')

```

Gambar 4. 16. Pengujian Nilai Gamma Sigmoid

```

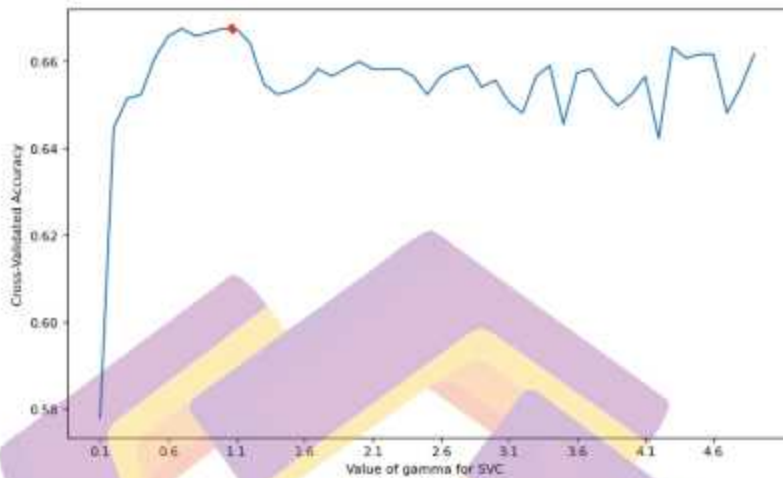
gamma_range=list(np.arange(0.1,5,0.1))
acc_score=[]
for g in gamma_range:
    svc = SVC(kernel='poly', gamma=g)
    scores = cross_val_score(svc, test_tf, dfnew['nilai'], cv=10, scoring='accuracy')
    acc_score.append(scores.mean())

figure(num=None, figsize=(10, 6), dpi=80, facecolor='w', edgecolor='k')
plt.plot(gamma_range, acc_score)
plt.xlabel('Value of gamma for SVC')
plt.xticks(np.arange(0.1,5,0.5))
plt.ylabel('Cross-Validated Accuracy')

```

Gambar 4. 17. Pengujian Nilai Gamma Polynomial

Dari pengujian nilai gamma yang telah dilakukan seperti pada gambar 4.15, gambar 4.16 dan gambar 4.17 didapatkan hasil yang disajikan dalam bentuk grafik. Dalam pembuatan grafik ini peneliti memanfaatkan library matplotlib. Adapun hasil pengujian dapat dilihat pada gambar 4.18, gambar 4.19 dan gambar 4.20.

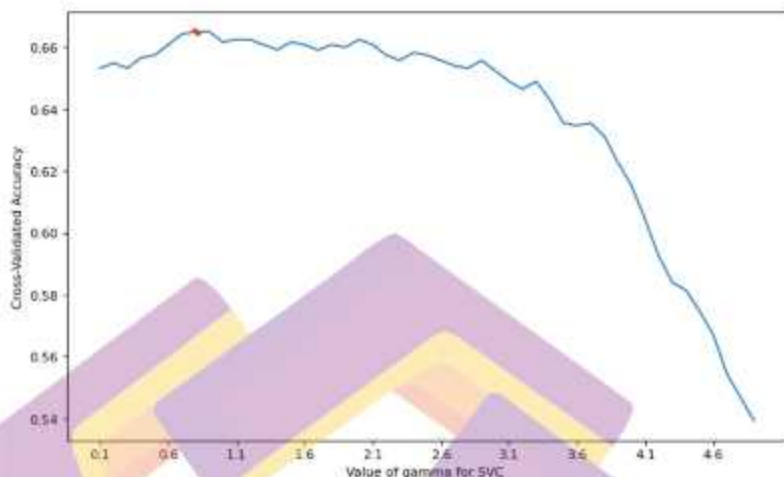


Gambar 4. 18. Hasil Pengujian Gamma RBF

Dari gambar 4.18 di atas dapat diketahui bahwa penentuan nilai gamma pada kernel RBF berpengaruh terhadap *cross validate accuracy*. Adapun peroleh *cross validate accuracy* dari setiap degree dapat dilihat pada tabel 4.6 berikut.

Tabel 4. 6. Penentuan Gamma RBF

No	Nilai Gamma	Cross Validate Accuracy
1	0,1	0,640
2	0,6	0,650
3	1,1	0,675
4	1,6	0,650
5	2,1	0,660
6	2,6	0,650
7	3,1	0,650
8	3,6	0,640
9	4,1	0,660
10	4,6	0,670
11	5,1	0,670

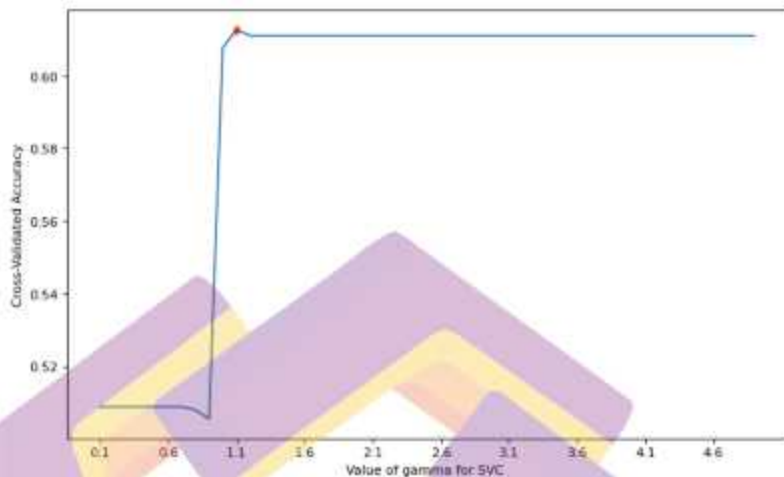


Gambar 4. 19. Hasil Pengujian Gamma Sigmoid

Dari gambar 4.19 di atas dapat diketahui bahwa penentuan nilai gamma pada kernel Sigmoid berpengaruh terhadap *cross validate accuracy*. Adapun peroleh *cross validate accuracy* dari setiap degree dapat dilihat pada tabel 4.7 berikut.

Tabel 4. 7 Penentuan Gamma Sigmoid

No	Nilai Gamma	Cross Validate Accuracy
1	0,1	0,65
2	0,8	0,665
3	1,1	0,66
4	1,6	0,66
5	2,1	0,66
6	2,6	0,65
7	3,1	0,65
8	3,6	0,64
9	4,1	0,58
10	4,6	0,56
11	5,1	0,54



Gambar 4. 20. Hasil Pengujian Gamma Polynomial

Dari gambar 4.20 di atas dapat diketahui bahwa penentuan nilai gamma pada kernel Polynomial berpengaruh terhadap *cross validate accuracy*. Adapun peroleh *cross validate accuracy* dari setiap degree dapat dilihat pada tabel 4.8 berikut.

Tabel 4. 8. Penentuan Gamma Polynomial

No	Nilai Gamma	Cross Validate Accuracy
1	0,1	0,51
2	0,6	0,51
3	1,1	0,615
4	1,6	0,61
5	2,1	0,61
6	2,6	0,61
7	3,1	0,61
8	3,6	0,61
9	4,1	0,61
10	4,6	0,61
11	5,1	0,61

Pada gambar 4.18, gambar 4.19, dan gambar 4.20 menyajikan nilai-nilai gamma pada sumbu x dan hasil akurasi pada sumbu y. Grafik tersebut memberikan gambaran visual bahwa nilai gamma yang digunakan pada kernel RBF, kernel sigmoid, dan kernel polynomial memiliki pengaruh terhadap hasil akurasi. Dari gambar 4.18, gambar 4.19, gambar 4.20 maka untuk penetapan nilai gamma yang akan digunakan untuk klasifikasi nanti akan disesuaikan untuk setiap kernelnya. Adapun nilai gamma tersebut dapat dilihat pada tabel 4.9 berikut.

Tabel 4.9. Nilai Gamma yang Dipilih

No	Kernel	Gamma	Cross Validate Accuracy
1	Polynomial	1,1	0,615
2	RBF	1,1	0,675
3	Sigmoid	0,8	0,665

Pada tabel 4.9 dapat diketahui bahwa nilai gamma yang akan digunakan untuk kernel RBF adalah 1,1 sedangkan untuk kernel sigmoid menggunakan nilai gamma 0,8, lalu untuk kernel polynomial menggunakan nilai gamma 1,1.

- d. Penentuan nilai itermax dilakukan dengan menguji nilai parameter 10,20,30,40,50,60,70,80,90,100. Nilai itermax yang digunakan adalah 100.
- e. Nilai toleransi konvergensi yang digunakan adalah 0,0001. Penggunaan parameter toleransi konvergensi berfungsi untuk mengendalikan kriteria berhenti dalam proses pelatihan SVM. Dalam implementasi SVM dengan menggunakan library yang sudah tersedia, seperti scikit-learn di Python.
- f. Coef 0 menggunakan nilai 1.0 karena konstanta akan digunakan sebagai pengaruh dalam pembentukan keputusan model.

4. Penerapan Kernel SVM

Tahap setelah nilai parameter yang akan digunakan untuk setiap kernel sudah didapatkan, maka peneliti melakukan pengujian dengan menggunakan *google colaboratory*. Implementasi kernel SVM akan dievaluasi performanya menggunakan *confusion matrix*. Adapun implementasi kernel SVM untuk klasifikasi sebagai berikut:

a. Implementasi Kernel Linear

Peneliti membuat objek *clf* yang merupakan *instance* dari kelas *SVC*. Pada kernel linear, *SVC* akan membangun garis pemisah linear untuk klasifikasi. Parameter yang digunakan peneliti pada implementasi kernel linear adalah parameter khusus, yaitu *C*, *tol*, dan *max_iter*. Setelah objek *SVC* diinisialisasi, metode *fit()* dipanggil untuk melatih model dengan menggunakan *Train_X_Tfidf* sebagai fitur dan *Train_Y* sebagai label. Metode *fit()* akan menghasilkan model yang telah dilatih. Setelah itu, peneliti menggunakan metode *cross_val_score* untuk melakukan validasi silang (*cross-validation*) menggunakan model *SVC* yang telah dilatih sebelumnya. Selanjutnya peneliti membuat fungsi untuk membagi data menjadi 100 *fold* (kelipatan) dan melatih model pada setiap *fold* secara bergantian. Fungsi aktivasi yang digunakan pada kernel linear $k_{x,y}=xTy+c$. Adapun *pseudocode* yang digunakan dapat dilihat pada gambar 4.21.


```

Fungsi klasifikasi_SVM_kernel_linear(data_latih, data_uji):
// Inisialisasi model SVM dengan parameter
model_SVM = inisialisasi_model_SVM()
// Melatih model dengan data latih dan labelnya menggunakan kernel linear
model_SVM.latih(data_latih, data_uji)

// Klasifikasi data uji menggunakan model SVM yang telah dilatih
label_prediksi = model_SVM.klasifikasi(data_uji)

// Mengevaluasi dengan confusion matrix
print(confusion_matrix(data_uji, label_prediksi))

```

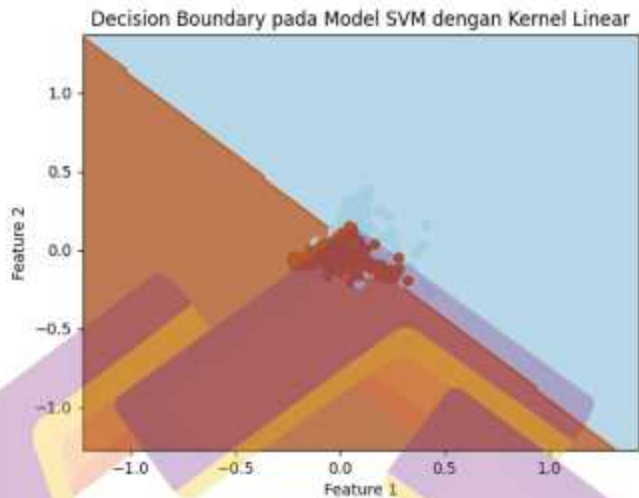
Gambar 4. 21. Pseudocode Kernel Linear

Pseudocode pada gambar 4.21 diterapkan untuk tiga skenario yaitu pembagian data latih dan data uji. Skenario 1 menggunakan data latih 80% dan data uji 20%. Skenario 2 menggunakan data latih 70% dan data uji 30%. Skenario 3 menggunakan data latih 60% dan data uji 40%. Pengujian dilakukan dengan *confusion matrix*. Hasil implementasi kernel SVM dapat dilihat pada tabel 4.10.

Tabel 4. 10. Hasil SVM Kernel Linear

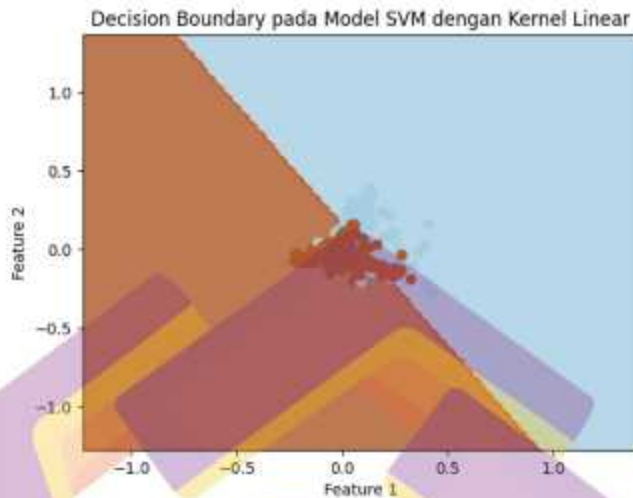
Confusion Matrix Kernel Linear	Skenario 1	Skenario 2	Skenario 3
Akurasi	73,24%	74,77%	76,48%
Recall	73,59%	75,40%	78,30%
Presisi	71,58%	69,70%	70,34%
F1-score	72,58%	72,44%	74,11%

Hasil klasifikasi menggunakan kernel linear pada SVM dapat divisualisasikan dalam bentuk grafik *decision boundary*. Adapun gambar *decision boundary* dapat dilihat pada gambar 4.22, gambar 4.23, dan gambar 4.24.



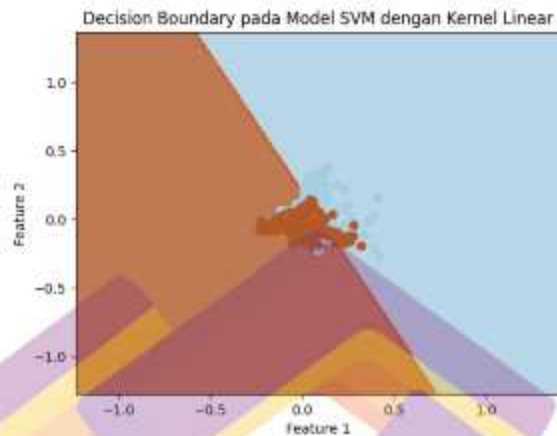
Gambar 4. 22. Decision Boundary Linear Skenario 1

Gambar 4.22 merupakan visualisasi klasifikasi menggunakan kernel linear. Pada sumbu x merepresentasikan feature 1, sedangkan sumbu y mewakili feature 2. Penggunaan warna biru dan coklat merupakan visualisasi dari data yang memiliki kelas yang berbeda. Garis pemisah dari kedua warna disebut decision boundary. Decision boundary menunjukkan bagaimana SVM memutuskan klasifikasi pada setiap titik dalam ruang fitur.



Gambar 4. 23. Decision Boundary Linear Skenario 2

Gambar 4.23 merupakan gambar klasifikasi menggunakan kernel linear. Terdapat dua sumbu yaitu sumbu x dan y yang mewakili feature. Pada sumbu x menggambarkan feature 1, sedangkan sumbu y mewakili feature 2. Penggunaan warna yang berbeda pada visualisasi karena untuk melihat garis pemisah (decision boundary). Decision boundary tersebut membagi area antara kelas berwarna coklat dan biru, menunjukkan bagaimana SVM memutuskan klasifikasi pada setiap titik dalam ruang fitur.



Gambar 4. 24. Decision Boundary Linear Skenario 3

Gambar 4.24 merupakan visualisasi klasifikasi menggunakan kernel linear. Pada gambar tersebut menjelaskan dua sumbu yaitu sumbu x dan y yang mewakili feature. Pada sumbu x menggambarkan feature 1, sedangkan sumbu y mewakili feature 2. Penggunaan warna yang berbeda pada visualisasi karena untuk melihat garis pemisah (decision boundary). Decision boundary tersebut membagi area antara kelas berwarna coklat dan biru, menunjukkan bagaimana SVM memutuskan klasifikasi pada setiap titik dalam ruang fitur.

b. Implementasi Kernel Polynomial

Peneliti membuat objek poly yang digunakan sebagai *instance* dari kelas SVC. Parameter yang digunakan yaitu *coef0*, *gamma*, dan *degree*. Parameter *coef0* diatur menjadi 1.0 yang berarti konstanta memiliki pengaruh dalam pembuatan keputusan model. *Gamma* digunakan untuk mengontrol seberapa sensitif model terhadap contoh pelatihan individual. Nilai 1.1 yang peneliti berikan menunjukkan

tingkat sensitivitas tertentu untuk fungsi kernel polynomial. Parameter *degree* mengatur derajat polinomial dalam kernel polynomial. Dalam hal ini, peneliti mengatur derajat polynomial menjadi 2, yang berarti model akan mempertimbangkan interaksi polynomial hingga derajat 2. Setelah objek SVC dengan kernel polynomial diinisialisasi, metode `fit()` dipanggil untuk melatih model menggunakan data `Train_X_Tfidf` sebagai fitur dan `Train_Y` sebagai label. Fungsi aktivasi yang digunakan pada kernel polynomial adalah $k[x,y] = [axTy+c]d$. *Pseudocode* yang digunakan peneliti dapat dilihat pada gambar 4.22.

```
Fungsi klasifikasi_SVM_kernel_Poly(data_latih, data_uji):
// Inisialisasi model SVM dengan parameter
model_SVM = inisialisasi_model_SVM()
// Melatih model dengan data latih dan labelnya menggunakan kernel polynomial
model_SVM.latih(data_latih, data_uji)

// Klasifikasi data uji menggunakan model SVM yang telah dilatih
label_prediksi = model_SVM.klasifikasi(data_uji)

// Mengevaluasi dengan confusion matrix
print(confusion_matrix(data_uji, label_prediksi))
```

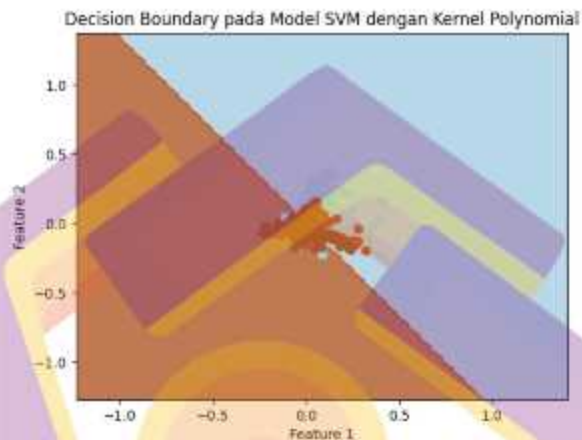
Gambar 4. 25. Pseudocode Kernel Polynomial

Pseudocode pada gambar 4.25 diterapkan untuk tiga skenario yaitu pembagian data latih dan data uji. Skenario 1 menggunakan data latih 80% dan data uji 20%. Skenario 2 menggunakan data latih 70% dan data uji 30%. Skenario 3 menggunakan data latih 60% dan data uji 40%. Pengujian dilakukan dengan *confusion matrix*. Hasil implementasi kernel SVM dapat dilihat pada tabel 4.11.

Tabel 4. 11. Hasil SVM Kernel Polynomial

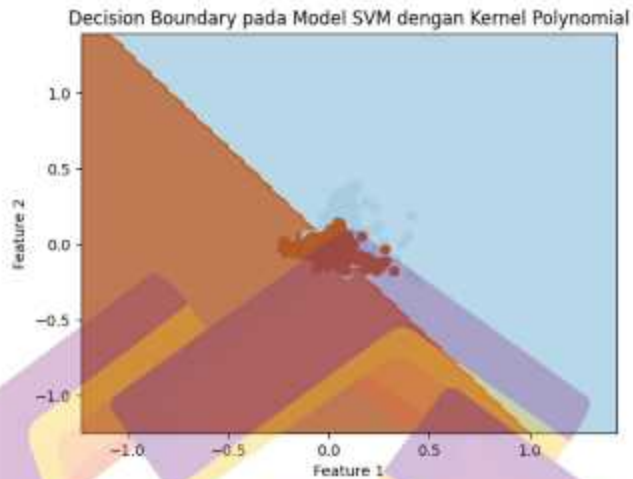
Confusion Matrix Kernel Polynomial	Skenario 1	Skenario 2	Skenario 3
Akurasi	86,49%	86,13%	82,70%
Recall	84,46%	84,50%	80,21%
Presisi	89,07%	86,74%	84,75%
F1-score	86,70%	85,61%	82,42%

Hasil klasifikasi menggunakan kernel polynomial pada SVM dapat divisualisasikan dalam bentuk grafik *decision boundary*. Adapun gambar *decision boundary* dapat dilihat pada gambar 4.26, gambar 4.27, dan gambar 4.28.



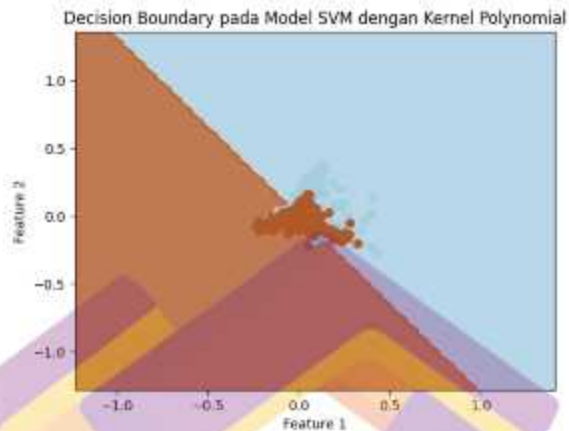
Gambar 4. 26. Decision Boundary Polynomial Skenario 1

Gambar 4.26 merupakan visualisasi klasifikasi menggunakan kernel polynomial. Terdapat dua sumbu yaitu sumbu x dan y yang mewakili feature. Pada sumbu x menggambarkan feature 1, sedangkan sumbu y mewakili feature 2. Penggunaan warna biru dan coklat merupakan visualisasi dari data yang memiliki kelas yang berbeda. Garis pemisah dari kedua warna disebut *decision boundary*. *Decision boundary* menunjukkan bagaimana SVM memutuskan klasifikasi pada setiap titik dalam ruang fitur.



Gambar 4. 27. Decision Boundary Polynomial Skenario 2

Gambar 4.27 merupakan visualisasi klasifikasi menggunakan kernel polynomial. Pada sumbu x menggambarkan feature 1, sedangkan sumbu y mewakili feature 2. Penggunaan warna yang berbeda pada visualisasi karena untuk melihat garis pemisah (decision boundary). Decision boundary tersebut membagi area antara kelas berwarna coklat dan biru, menunjukkan bagaimana SVM memutuskan klasifikasi pada setiap titik dalam ruang fitur.



Gambar 4. 28. Decision Boundary Polynomial Skenario 3

Gambar 4.28 merupakan visualisasi klasifikasi menggunakan kernel polynomial. Pada sumbu x menggambarkan feature 1, sedangkan sumbu y mewakili feature 2. Penggunaan warna yang berbeda pada visualisasi karena untuk melihat garis pemisah (decision boundary). Decision boundary tersebut membagi area antara kelas berwarna coklat dan biru, menunjukkan bagaimana SVM memutuskan klasifikasi pada setiap titik dalam ruang fitur.

c. Implementasi Kernel RBF

Peneliti membuat objek rbf yang merupakan *instance* dari kelas SVC dari *library* Scikit-learn. Parameter gamma diatur dengan nilai 0.8, yang mengontrol sejauh apa pengaruh dari sebuah sampel pelatihan dapat mencapai dalam pembentukan batas keputusan. Nilai gamma yang lebih tinggi menghasilkan pengaruh yang lebih besar dan kompleksitas model yang lebih tinggi. Proses

pelatihan SVM menggunakan metode fit. `Train_X_Tfidf` adalah matriks fitur yang berisi vektor representasi fitur dari data pelatihan, sedangkan `Train_Y` adalah array target yang berisi label kelas dari data pelatihan yang sesuai. Model SVM akan belajar dari data ini untuk membentuk batas keputusan yang memisahkan kelas-kelas yang berbeda.

Peneliti melakukan validasi silang pada model SVM rbf menggunakan fungsi `cross_val_score` dari pustaka `Scikit-learn`. Parameter pertama adalah model yang akan divalidasi (rbf), `Train_X_Tfidf` adalah matriks fitur data pelatihan, `Train_Y` adalah array target yang berisi label kelas data pelatihan, dan `cv=100` menentukan jumlah lipatan dalam validasi silang (dalam hal ini 100 lipatan). Fungsi ini akan mengembalikan nilai akurasi dari setiap lipatan validasi sebagai array scores. Hasil prediksi akan disimpan dalam array `y_pred`. Fungsi aktivasi yang digunakan pada kernel polynomial adalah $k[x,y] = \exp[-y\|x-y\|^2]$. Adapun *pseudocode* yang digunakan peneliti dapat dilihat pada gambar 4.23 berikut.

```
Fungsi klasifikasi_SVM_kernel_RBF(data_latih, data_uji):
// Inisialisasi model SVM dengan parameter
model_SVM = inisialisasi_model_SVM()
// Melatih model dengan data latih dan labelnya menggunakan kernel RBF
model_SVM.latih(data_latih, data_uji)
// Klasifikasi data uji menggunakan model SVM yang telah dilatih
label_prediksi = model_SVM.klasifikasi(data_uji)
// Mengevaluasi dengan confusion matrix
print(confusion_matrix(data_uji, label_prediksi))
```

Gambar 4. 29. Pseudocode Kernel RBF

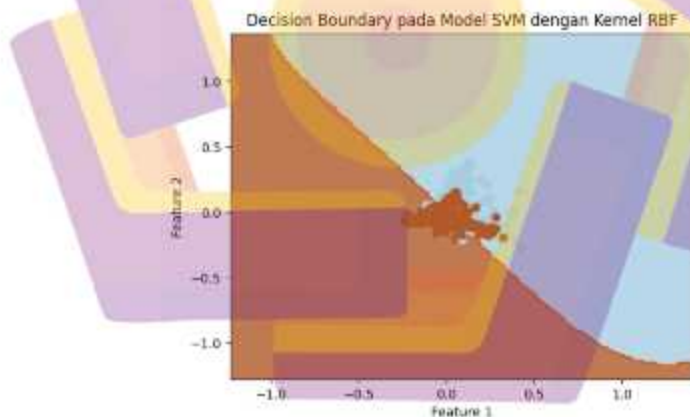
Pseudocode pada gambar 4.29 diterapkan untuk tiga skenario yaitu pembagian data latih dan data uji. Skenario 1 menggunakan data latih 80% dan data

uji 20%. Skenario 2 menggunakan data latih 70% dan data uji 30%. Skenario 3 menggunakan data latih 60% dan data uji 40%. Pengujian dilakukan dengan *confusion matrix*. Hasil implementasi kernel SVM dapat dilihat pada tabel 4.12.

Tabel 4. 12. Hasil SVM Kernel RBF

Confusion Matrix Kernel RBF	Skenario 1	Skenario 2	Skenario 3
Akurasi	85,95%	85,23%	83,51%
Recall	84,30%	82,5%	79,29%
Presi	87,98%	87,5%	88,70%
F1-score	86,10%	84,93%	83,73%

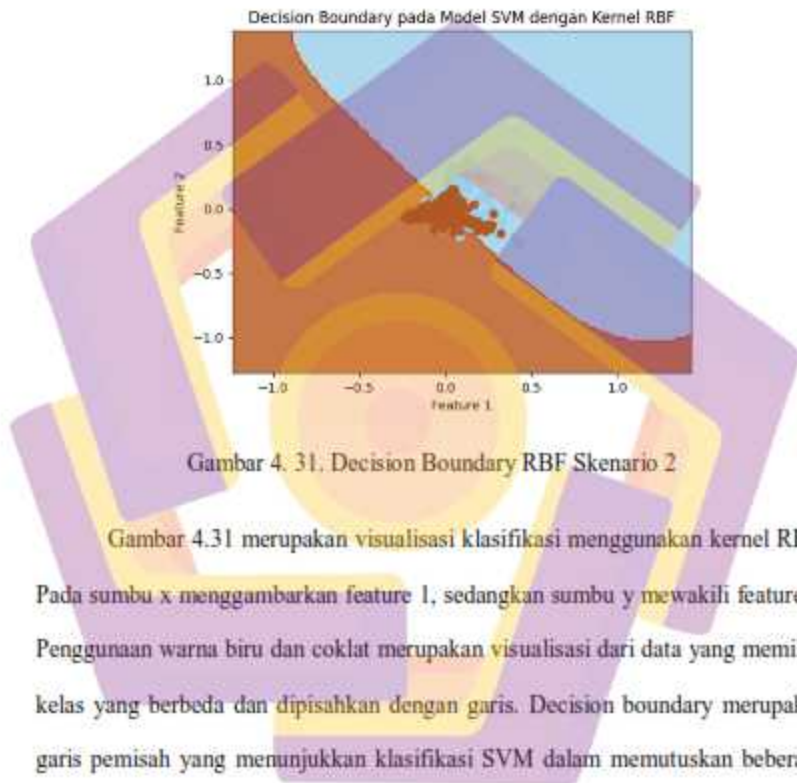
Hasil klasifikasi menggunakan kernel RBF pada SVM dapat divisualisasikan dalam bentuk grafik *decision boundary*. Adapun gambar *decision boundary* dapat dilihat pada gambar 4.30, gambar 4.31, dan gambar 4.32.



Gambar 4. 30. Decision Boundary RBF Skenario 1

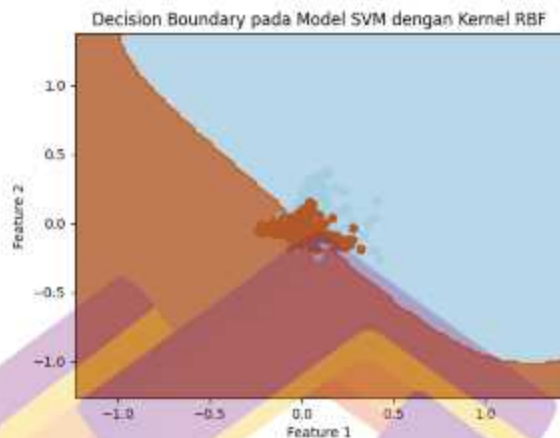
Gambar 4.30 merupakan visualisasi klasifikasi menggunakan kernel RBF. Terdapat dua sumbu yaitu sumbu x dan y yang mewakili feature, Pada sumbu x menggambarkan feature 1, sedangkan sumbu y mewakili feature 2. Penggunaan

warna biru dan coklat merupakan visualisasi dari data yang memiliki kelas yang berbeda dan dipisahkan dengan garis. Decision boundary merupakan garis pemisah yang menunjukkan klasifikasi SVM dalam memutuskan beberapa titik.



Gambar 4. 31. Decision Boundary RBF Skenario 2

Gambar 4.31 merupakan visualisasi klasifikasi menggunakan kernel RBF. Pada sumbu x menggambarkan feature 1, sedangkan sumbu y mewakili feature 2. Penggunaan warna biru dan coklat merupakan visualisasi dari data yang memiliki kelas yang berbeda dan dipisahkan dengan garis. Decision boundary merupakan garis pemisah yang menunjukkan klasifikasi SVM dalam memutuskan beberapa titik.



Gambar 4. 32. Decision Boundary RBF Skenario 3

Gambar 4.32 merupakan visualisasi dua sumbu dalam klasifikasi menggunakan kernel RBF. Pada sumbu x menggambarkan feature 1, sedangkan sumbu y mewakili feature 2. Penggunaan warna biru dan coklat merupakan visualisasi dari data yang memiliki kelas yang berbeda dan dipisahkan dengan garis. Decision boundary merupakan garis pemisah yang menunjukkan klasifikasi SVM dalam memutuskan beberapa titik.

d. Implementasi Kernel Sigmoid

Peneliti membuat objek sig yang merupakan instance dari kelas SVC (Support Vector Classifier) dari pustaka Scikit-learn. Parameter gamma diatur dengan nilai 1, yang mengontrol sejauh apa pengaruh dari sebuah sampel pelatihan dapat mencapai dalam pembentukan batas keputusan. Parameter coef0 diatur dengan nilai 0.8, yang merupakan parameter tambahan dalam kernel sigmoid. Peneliti menambahkan fungsi cross_val_score dari pustaka Scikit-learn. Parameter

pertama adalah model yang akan divalidasi (sig), Train_X_Tfidf adalah matriks fitur data pelatihan, Train_Y adalah array target yang berisi label kelas data pelatihan, dan cv=100 menentukan jumlah lipatan dalam validasi silang (dalam hal ini 100 lipatan). Fungsi ini akan mengembalikan nilai akurasi dari setiap lipatan validasi sebagai array scores. Fungsi aktivasi yang digunakan pada kernel polynomial adalah $k[x,y]=\tanh [axTy+c]$. *Pseudocode* yang digunakan peneliti dapat dilihat pada gambar 4.33 berikut.

```

Fungsi klasifikasi_SVM_kernel_Sigmoid(data_latih, data_uji):
// Inisialisasi model SVM dengan parameter
model_SVM = inisialisasi_model_SVM()
// Melatih model dengan data latih dan labelnya menggunakan kernel Sigmoid
model_SVM.latih(data_latih, data_uji)

// Klasifikasi data uji menggunakan model SVM yang telah dilatih
label_prediksi = model_SVM.klasifikasi(data_uji)

// Mengevaluasi dengan confusion matrix
print(confusion_matrix(data_uji, label_prediksi))

```

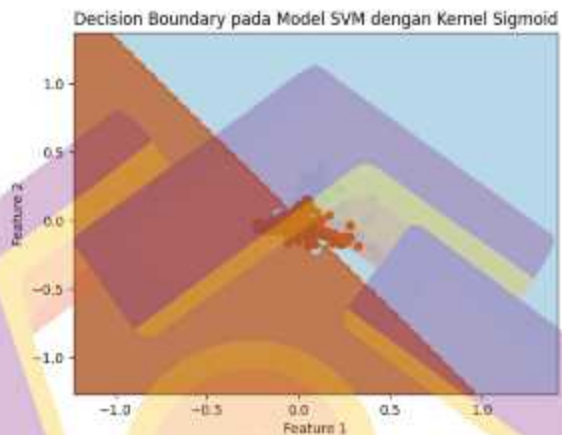
Gambar 4. 33. Pseudocode Kernel Sigmoid

Pseudocode pada gambar 4.33 diterapkan untuk tiga skenario yaitu pembagian data latih dan data uji. Skenario 1 menggunakan data latih 80% dan data uji 20%. Skenario 2 menggunakan data latih 70% dan data uji 30%. Skenario 3 menggunakan data latih 60% dan data uji 40%. Pengujian dilakukan dengan *confusion matrix*. Hasil implementasi kernel SVM dapat dilihat pada tabel 4.13.

Tabel 4.13. Hasil SVM Kernel Sigmoid

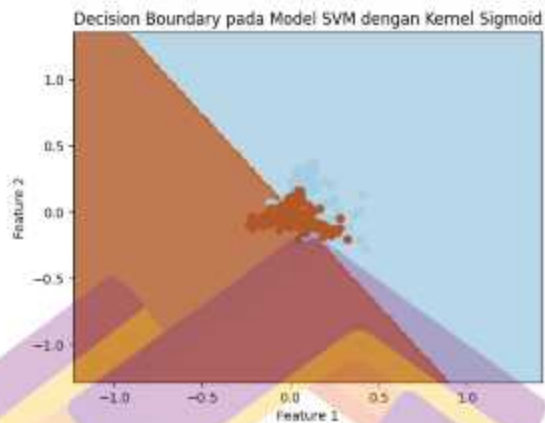
CF Kernel Sigmoid	Skenario 1	Skenario 2	Skenario 3
Akurasi	84,59%	84,50%	80,67%
Recall	82,81%	83,21%	77,40%
Presisi	86,89%	84,47%	84,18%
F1-score	84,80%	83,83%	80,65%

Hasil klasifikasi menggunakan kernel Sigmoid pada SVM dapat divisualisasikan dalam bentuk grafik *decision boundary*. Adapun gambar *decision boundary* dapat dilihat pada gambar 4.34, gambar 4.35, dan gambar 4.36.



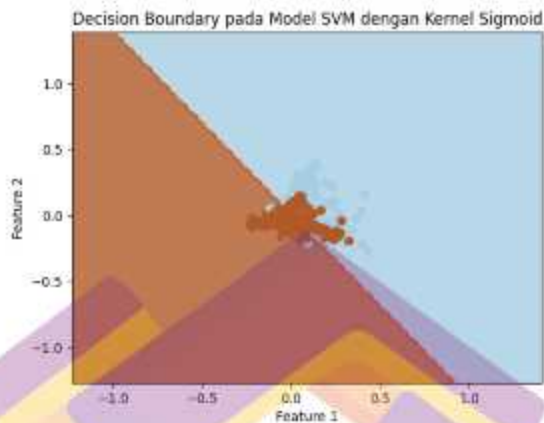
Gambar 4. 34. Decision Boundary Sigmoid Skenario 1

Gambar 4.34 merupakan visualisasi klasifikasi menggunakan kernel sigmoid. Terdapat dua sumbu yaitu sumbu x dan y yang mewakili feature. Pada sumbu x menggambarkan feature 1, sedangkan sumbu y mewakili feature 2. Penggunaan warna biru dan coklat merupakan visualisasi dari data yang memiliki kelas yang berbeda. Garis pemisah dari kedua warna disebut *decision boundary*. *Decision boundary* menunjukkan bagaimana SVM memutuskan klasifikasi pada setiap titik dalam ruang fitur.



Gambar 4.35. Decision Boundary Sigmoid Skenario 2

Gambar 4.35 merupakan visualisasi dua sumbu dalam klasifikasi menggunakan kernel sigmoid. Pada sumbu x menggambarkan feature 1, sedangkan sumbu y mewakili feature 2. Penggunaan warna biru dan coklat merupakan visualisasi dari data yang memiliki kelas yang berbeda dan dipisahkan dengan garis. Decision boundary merupakan garis pemisah yang menunjukkan klasifikasi SVM dalam memutuskan beberapa titik.



Gambar 4.36. Decision Boundary Sigmoid Skenario 3

Gambar 4.36 merupakan visualisasi klasifikasi menggunakan kernel sigmoid. Pada sumbu x merepresentasikan feature 1, sedangkan sumbu y mewakili feature 2. Penggunaan warna biru dan coklat merupakan visualisasi dari data yang memiliki kelas yang berbeda. Garis pemisah dari kedua warna disebut decision boundary. Decision boundary menunjukkan bagaimana SVM memutuskan klasifikasi pada setiap titik dalam ruang fitur.

5. Analisis Perbandingan Kernel dan Parameter SVM

Dari tabel 4.10, tabel 4.11, tabel 4.12, dan tabel 4.13 dapat dipermudah untuk membandingkan performa dengan melihat tabel 4.14 berikut.

Tabel 4. 14. Perbandingan Kernel SVM

Kernel	Confusion Matrix	Skenario 1	Skenario 2	Skenario 3
Linear	Akurasi	73,24%	74,77%	76,48%
	Recall	73,59%	75,40%	78,30%
	Presisi	71,58%	69,70%	70,34%
	F1-score	72,58%	72,44%	74,11%
Polynomial	Akurasi	86,49%	86,13%	82,70%
	Recall	84,46%	84,50%	80,21%
	Presisi	89,07%	86,74%	84,75%
	F1-score	86,70%	85,61%	82,42%
RBF	Akurasi	85,95%	85,23%	83,51%
	Recall	84,30%	82,5%	79,29%
	Presisi	87,98%	87,5%	88,70%
	F1-score	86,10%	84,93%	83,73%
Sigmoid	Akurasi	84,59%	84,50%	80,67%
	Recall	82,81%	83,21%	77,40%
	Presisi	86,89%	84,47%	84,18%
	F1-score	84,80%	83,83%	80,65%

Dari tabel 4.14 diatas dapat diketahui bahwa kernel yang memiliki performa baik jika dilihat dari performa akurasi, presisi, recall, dan f1-score yaitu kernel Polynomial. Pada penelitian ini peneliti akan menetapkan kernel terbaik berdasarkan nilai akurasi, sehingga kernel yang akan dipilih menjadi kernel terbaik adalah kernel polynomial.

Hasil penelitian ini menjadi lebih baik dari penelitian sebelumnya karena mengikuti saran dari (Muh.Fitra, 2021) yaitu untuk menambah jumlah dataset. Penelitian (Muh.Fitra, 2021) menghasilkan akurasi 70%, sedangkan pada penelitian ini akurasi tertinggi yang didapat yaitu 86,49%. Kernel sigmoid pada penelitian ini memiliki nilai presisi yang lebih unggul dibanding penelitian sebelumnya yang hanya mendapat presisi 75,1% (Christevan Destisus, dkk, 2020).

Faktor yang mempengaruhi hasil pengujian yang telah dilakukan yaitu jenis kernel yang digunakan seperti dapat dilihat pada tabel 4.14 di atas. Selain itu penggunaan parameter dan nilai parameter (konstanta). Parameter yang digunakan pada penelitian ini merupakan parameter khusus untuk tiap kernel (parameter asli dari kernel). Konstanta yang digunakan yaitu berdasarkan pengujian *range* nilai masing-masing parameter. Dari beberapa kali pengujian yang dilakukan peneliti, faktor yang paling menentukan hasil adalah penetapan nilai konstanta dan penggunaan parameter. Contoh uji coba yang dilakukan peneliti dapat dilihat pada tabel 4.15 berikut.

Tabel 4. 15, Uji Coba Pengaruh Parameter

Kernel	Jumlah Data	Parameter Gamma	Akurasi	Presisi	Recall	F1 Score
RBF	1850	auto	48,92%	100%	48,92%	65,70%
		1.1	85,94%	87,97%	84,29%	86,10%

Dari tabel 4.15 dapat menjadi contoh salah satu uji coba yang telah dilakukan peneliti bahwa penggunaan parameter dan penentuan nilai parameter dapat mempengaruhi hasil performa. Penggunaan parameter gamma yang

didefinisikan nilai parameternya lebih baik daripada penggunaan parameter γ auto. Hal ini juga berlaku pada parameter-parameter lainnya.

Pengujian pada penelitian ini melibatkan nilai akurasi, recall, f_1 -score, dan presisi. Akan tetapi untuk penentuan kernel terbaik akan dipilih berdasarkan akurasi yang diperoleh. Hal ini dikarenakan kurasi mengukur sejauh mana model berhasil memprediksi dengan benar seluruh kelas secara keseluruhan. Sehingga pengujian presisi, recall, dan f_1 -score hanya sebagai pendukung pengujian. Presisi dalam penelitian ini digunakan untuk mengukur seberapa banyak dari prediksi positif yang sebenarnya benar untuk menentukan tweet cyberbullying. Pengujian menggunakan recall mengukur sejauh mana model berhasil menemukan semua kasus positif yang sebenarnya. Sedangkan F_1 -score digunakan untuk menguji keseimbangan antara presisi dan recall.

BAB V PENUTUP

5.1. Kesimpulan

Kesimpulan yang didapat pada penelitian ini adalah:

- a. SVM dapat melakukan klasifikasi cyberbullying berbahasa indonesia dengan kernel-kernelnya. Cara yang dilakukan peneliti adalah dengan mengumpulkan dataset dari twitter, melakukan preprocessing, ekstraksi fitur dengan TF-IDF, mengimplementasikan setiap kernel yang dipilih dengan memperhatikan parameter. Pilihan kernel linear, polynomial, radial RBF, atau sigmoid, berpengaruh terhadap bagaimana SVM dapat menangani hubungan non-linear antar data.
- b. Hasil akurasi yang didapat kernel linear pada skenario 1 memperoleh akurasi 73,24%, skenario 2 memperoleh akurasi 74,77%, dan skenario 3 memperoleh akurasi 76,48%. Hasil akurasi yang didapat kernel polynomial pada skenario 1 memperoleh akurasi 86,49%, skenario 2 memperoleh akurasi 86,13%, dan skenario 3 memperoleh akurasi 82,70%. Hasil akurasi yang didapat kernel RBF pada skenario 1 memperoleh akurasi 86,95%, skenario 2 memperoleh akurasi 85,23%, dan skenario 3 memperoleh akurasi 83,51%. Hasil akurasi yang didapat kernel polynomial pada skenario 1 memperoleh akurasi 86,49%, skenario 2 memperoleh akurasi 86,13%, dan skenario 3 memperoleh akurasi 82,70%. Hasil akurasi yang didapat kernel sigmoid pada skenario 1 memperoleh akurasi 84,59 %, skenario 2 memperoleh akurasi 84,50%, dan skenario 3 memperoleh akurasi 80,67%.

Perolehan nilai akurasi dari beberapa kernel tersebut, menunjukkan bahwa ketika melakukan klasifikasi perlu untuk melakukan percobaan dengan berbagai kernel dan parameter untuk menyesuaikan hyperparameter, dan melakukan validasi silang untuk menentukan kernel mana yang memberikan kinerja terbaik.

- c. Dari empat kernel yang digunakan dapat diketahui bahwa kernel yang memiliki akurasi terbaik adalah Polynomial. Kernel ini memiliki akurasi tertinggi pada 3 skenario dibandingkan dengan kernel lainnya. Penentuan kernel terbaik ini bergantung pada jenis data yang digunakan dan konstanta dari tiap parameter.

5.2.Saran

Adapun saran untuk peneliti selanjutnya sebagai berikut:

- a. Pada penelitian ini, peneliti hanya menggunakan parameter wajib untuk setiap kernel yang diuji dengan range nilai yang masih sedikit. Untuk penelitian selanjutnya, peneliti diharapkan dapat mencoba beberapa nilai pada setiap parameter dengan range yang beragam. Sehingga mendapat nilai parameter yang optimal dan diharapkan dapat menaikkan performa akurasi.
- b. Dataset yang digunakan perlu ditambah untuk meningkatkan validasi pemodelan. Peneliti selanjutnya dapat menambahkan kata kunci selain dari yang sudah ditulis pada penelitian ini.

DAFTAR PUSTAKA

PUSTAKA BUKU

- Ahmed, A.Abd Latif, Brij, Gupta., Deepak, Gupta., Dragan, Perakovic., 2021, Data Mining Approaches for Big Data and Sentiment Analysis in Social Media, IGI Global
- Fajar Junaedi, 2020, Etika Komunikasi di Era Siber: Teori dan Praktik, PT RajaGrafindo Persada, Depok
- Fikri Aldi Nugraha, Nisa Hanum Harani, Roni Habibi, 2020, Analisis Sentimen Terhadap Pembatasan Sosial Menggunakan Deep Learning, Kreatif Industri Nusantara, Bandung
- Eva, Damila., Mauliza., Mutammimul, Ula., 2019, Aplikasi Teknologi Sistem Pakar Berbasis Fuzzy Clustering, Yayasan Kita Menulis
- Onno, W.Purbo., 2019, Text Mining - Analisis MedSos, Kekustan Brand & Intelegen di Internet, Andi Publisher, Yogyakarta
- Rahmadya Trias Hendaryanto, Herlawati, 2020, Data Mining dan Machine Learning Menggunakan Matlab dan Python, Informatika, Bandung
- Primartha, R., 2021, Algoritma Machine Learning, Informatika, Bandung

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

- Dewi, F. dkk., 2022, Penerapan Binary Particle Swarm OptimizationSupport Vector Machineuntuk Klasifikasi Komentar Cyberbullyingdi Instagram, Xplore: Journal of Statistics, e-ISSN: 2655-2744, 2022
- I, Putu Gede Hendra Suputra, dkk., 2019, Perbandingan TF terhadap Hasil Evaluasi Information Retrieval, Jurnal Elektronik Ilmu Komputer Udayana, e-ISSN: 2654-5101, p-ISSN: 2301-5373
- Said, R.K.Bahasyim, dkk., 2021, Otomasi Penelusuran Kebutuhan ke Kode Program menggunakan TF-IDF, e-Proceeding of Engineering: Vol.8 Issue.2 April, 2021

PUSTAKA ELEKTRONIK

- Alpaydin, Ethem, Introduction to Machine Learning, The MIT Press, ISBN 978-0-262-02818-9,

https://www.google.co.id/books/edition/Introduction_to_Machine_Learning/NP5bBAAAQBAJ?hl=en&gbpv=1&dq=machine+learning+adalah&printsec=frontcover, diakses 08 Oktober 2022 pukul 10.30

Almurity, Samar.; Fattah, M.A., Arabic CyberBullying Detection Using Arabic Sentiment Analysis, ISSN 2356-82016, https://ejle.journals.ekb.eg/article_160441.html, diakses 02 Oktober 2022 pukul 20.33

Ardani, Muhammad.; Nasir, Muhammad, Data Mining Algoritma dan Implementasi, Penerbit Andi, ISBN 978-623-01-0725-2 https://www.google.co.id/books/edition/Data_Mining_Algoritma_dan_Implementasi/AtcCEAAAQBAJ?hl=en&gbpv=1&dq=data+mining+adalah&printsec=frontcover, diakses 08 Oktober 2022 pukul 21.30

Aulia, T.M.P.; Arifin, N.; Mayasari, R., Perbandingan Kernel Support Vector Machine (SVM) dalam Penerapan Analisis Sentimen Vaksinasi Covid-19, ISSN 2598-7305, E-ISSN 2598-9642, <https://jurnal.instiki.ac.id/index.php/sintechjournal/article/view/762>, diakses 02 Oktober 2022 pukul 19.45

Destisus, Christevan.; Wella; Suryasari., Support Vector Machine VS Information Gain: Analisis Sentimen Cyberbullying di Twitter Indonesia, ISSN 2085-4579, <https://ejournals.umn.ac.id/index.php/SI/article/view/1740>, diakses 02 Oktober 2022 pukul 09.44

Fitriani, K.; Isbandi; Amaliyah, A., Perancangan Sistem Manajemen dengan Menggunakan Metode Text Mining di Kantor Kelurahan Sekejati, ISSN 2088 9135, <http://ojs.uninus.ac.id/index.php/Telematika/article/view/2125/1220>, diakses 08 Oktober 2022 pukul 13.27

<https://www.unicef.org/indonesia/id/child-protection/apa-itu-cyberbullying> diakses 01 Oktober 2022

Ige, Tosin.; Adewale, Sikiru., AI Powered Anti-Cyber Bullying System using Machine Learning Algorithm of Multinomial Naïve Bayes and Optimized Linear Support Vector Machine, <https://arxiv.org/abs/2207.11897>, diakses 02 Oktober 2022 pukul 09.22

Karyanti; Aminudin, Cyberbullying & Body Shaming, ISBN 978-602-451-335-1, https://www.google.co.id/books/edition/CYBERBULLYING_BODY_SHAMING/c840EAAAQBAJ?hl=en&gbpv=1&dq=cyberbullying&printsec=frontcover, diakses 04 Oktober 2022 pukul 07.37

Jo; Taeho, Machine Learning Foundations, ISBN 978-3-030-65899-1, https://www.google.co.id/books/edition/Machine_Learning_Foundations/0egdEA

AAQBAJ?hl=en&gbpv=1&dq=machine+learning+adalah&printsec=frontcover, diakses 08 Oktober 2022 pukul 10.22

- Juju, Dominikus.; Matamaya, Twitter, ISBN 978-979-27-5418-6 <https://www.google.co.id/books/edition/Twitter/-4xKDwAAQBAJ?hl=en&gbpv=1&dq=twitter+adalah&pg=PR10&printsec=frontcover>, diakses 08 Oktober 2022 pukul 09.55
- Kowalczyk, A. (2014). Linear Kernel: Why is it recommended for text classification ?, <https://www.svm-tutorial.com/2014/10/svm-linear-kernel-good-text-classification/> , diakses 14 November 2022 pada 11.20
- Kusuma, Danu Purba, Machine Learning Teori, Program, Dan Studi Kasus, ISBN 978-623-02-1083-9, https://www.google.co.id/books/edition/Machine_Learning_Teori_Program_Dan_Studi/4k3sDwAAQBAJ?hl=en&gbpv=1&dq=machine+learning+adalah&printsec=frontcover, diakses 09 Oktober 2022 pada 11.40
- Li, Tao.; dkk., Music Data Mining, ISBN 978-1-4398-3552-4, https://www.google.co.id/books/edition/Music_Data_Mining/_zc3vKDLUNIC?hl=en&gbpv=1&dq=data+mining+adalah&printsec=frontcover, diakses 08 Oktober 2022 pukul 10.30
- Marisa, Fitri.; dkk, Data Mining Konsep dan Penerapannya, E-ISBN 978-623-02-3818-5, https://www.google.co.id/books/edition/Data_Mining_Konsep_Dan_Penerapannya/Bt1VEAAAQBAJ?hl=en&gbpv=1&dq=data+mining+adalah&printsec=frontcover, diakses 08 Oktober 2022 pada 14.25
- Nigrum, H.C.S., 10 Oktober 2022, Perbandingan Metode Support Vector Machine (SVM) Linear, Radial Basic Function (RBF), dan Polynomial Kernel dalam Klasifikasi Bidang Studi Lanjut Pilihan Alumni UII, <https://dspace.uui.ac.id/handle/123456789/7791>
- Normawati, Dwi; Prayogi, S.A., Implementasi Naive Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter, EISSN: 2549-7200, <https://ejournal.tunasbangsa.ac.id/index.php/jsakti/article/view/369/348>, diakses 08 Oktober 2022 pada 18.45
- Provost, F., Kohavi, R. Guest Editors' Introduction: On Applied Research in Machine Learning. Machine Learning 30, 127-132 (1998), <https://doi.org/10.1023/A:1007442505281>, diakses 15 November 2022 pada 14.20
- Nugraha, Fikri Aldi, Analisis Sentimen Terhadap Pembatasan Sosial Menggunakan Deep Learning, ISBN 070-623-7898-64-1, https://www.google.co.id/books/edition/Analisis_Sentimen_Terhadap_Pembatasan

n_So/f738DwAAQBAJ?hl=en&gbpv=1&dq=analisis+sentimen&pg=PA25&printsec=frontcover, diakses 08 Oktober 2022 pada 15.48

- Patel, S. (2017, Mei 3). Chapter 2 : SVM (Support Vector Machine) — Theory. 14 November 2022, Machine Learning 101: <https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72>
- Prabowo, W.A.; Azizah, F, Sentiment Analysis for Detecting Cyberbullying Using TF-IDF and SVM, ISSN Electronic Media: 2580-0760, <https://www.jurnal.iaii.or.id/index.php/RESTI/article/view/2753>, diakses 02 Oktober 2022 pada 10.15
- Pratiwi, B.P; Handayani, A.S; Sarjana., Pengukuran Kinerja Sistem Kualitas Udara Dengan Teknologi WSN Menggunakan Confusion Matrix, E-ISSN 2460-4801/2447-6645, <http://journal.upgris.ac.id/index.php/JIU/article/view/6552/3859>, diakses 09 Oktober 2022 pukul 08.22
- Pratiwi, Dinda Ayu.; dkk., Seleksi Calon Kelulusan Tepat Waktu Mahasiswa Teknik Informatika Menggunakan Metode Naive Bayes, ISBN 978-623-6762-01-1, https://www.google.co.id/books/edition/SELEKSI_CALON_KELULUSAN_TEPAT_WAKTU_MAHAD6wGEAAAQBAJ?hl=en&gbpv=1&dq=svm+adalah&pg=PA47&printsec=frontcover, diakses 09 Oktober 2022 pada 10.18
- Putri, A.; Muzakir, A., Analisis Sentimen *Cyberbullying* KPOP di Media Sosial Twitter Menggunakan Metode Naive Bayes, e-ISSN: 2548-1398, <https://www.jurnal.syntaxliterate.co.id/index.php/syntax-literate/article/view/9334/5314>, diakses 02 Oktober 2022 pada 19.44
- Raudhoti, L.A.H.; Herdiani.; Romadhony., Identifikasi Cyberbullying pada Kolom Komentar Instagram dengan Metode Support Vector Machine dan Semantic Similarity, E-ISSN:2541-0806, <https://jcosine.if.unram.ac.id/index.php/jcosine/article/view/318>, diakses 02 Oktober 2022 pukul 21.20
- Raja, Rohit.; dkk., Data Mining and Machine Learning Application, ISBN 978-1-119-79178-2, https://www.google.co.id/books/edition/Data_Mining_and_Machine_Learning_Application/b8d6EAAAQBAJ?hl=en&gbpv=1&dq=data+mining+adalah&printsec=frontcover, diakses 08 Oktober 2022 pukul 21.42
- Rangkuti, Yulita Molig.; dkk, Pengantar Pemrograman Python, ISBN 978-623-154-0, https://www.google.co.id/books/edition/Pengantar_Pemrograman_Python/2fLEAAAQBAJ?hl=en&gbpv=1&dq=python+adalah&pg=PA246&printsec=frontcover, diakses 09 Oktober 2022 pukul 20.14
- Rianti, D.L.; Umaidah, Y.; Voutama, A., Tren Marketplace Berdasarkan Klasifikasi Ulasan Pelanggan Menggunakan Perbandingan Kernel Support Vector Machine,

e-ISSN: 2549 -2837,

<https://journal.lppmunindra.ac.id/index.php/STRING/article/view/9993>, diakses 02 Oktober 2022 pukul 16.18

Rizki, M.F.; dkk., Analisis Sentiment Cyberbullying pada Sosial Media Twitter Menggunakan Metode Support Vector Machine, <https://doi.org/10.36040/jati.v5i2.3808>,

<https://ejournal.itn.ac.id/index.php/jati/article/view/3808/2852>, diakses 08 Oktober 2022 pukul 15.28

Safitri, R.; dkk, Analisis Sentimen: Metode Alternatif Metode Alternatif Penelitian Big Data, ISBN

-978-623-296-280- 4,

https://www.google.co.id/books/edition/Analisis_Sentimen/TVdmEAAAQBAJ?hl=en&gbpv=1&dq=analisis+sentimen&printsec=frontcover, diakses 08 Oktober 2022 pukul 17.10

Tahyudin, Imam, Pengenalan Machine Learning Menggunakan Jupyter Notebook, ISBN 978-602-50408-4-9,

https://www.google.co.id/books/edition/Pengenalan_Machine_Learning_Menggunakan/_uMREAAAQBAJ?hl=en&gbpv=1&dq=machine+learning+adalah&printsec=frontcover, diakses 09 Oktober 2022 pukul 19.16

Wardana, Belajar Pemrograman dan Hacking Menggunakan Python, ISBN 978-6230-01-050-7,

https://www.google.co.id/books/edition/Belajar_Pemrograman_dan_Hacking_Menggunakan/kCvGDwAAQBAJ?hl=en&gbpv=1&dq=python+adalah&pg=PA2&printsec=frontcover, diakses 09 Oktober 2022, pukul 13.49

Yudhanto, Y.; dkk, Belajar Mengelola Facebook dan Twitter, https://www.google.co.id/books/edition/Belajar_Mengelola_Facebook_dan_Twitter/vJtQDwAAQBAJ?hl=en&gbpv=1&dq=twitter+adalah&pg=PA129&printsec=frontcover,

diakses 08 Oktober 2022 pada 14.52