

**ANALISIS SENTIMEN MASYARAKAT TERHADAP KEBIJAKAN
PPKM DENGAN MENERAPKAN *OPPOSITION-BASED WHALE
OPTIMIZATION ALGORITHM* PADA KLASIFIKASI SVM**



SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan
Pendidikan Diploma Empat (D-4) Program Studi Teknik Komputer dan Jaringan
Jurusan Teknik Elektro
Politeknik Negeri Ujung Pandang

ASRUL SAID

42518058

**PROGRAM STUDI D4 TEKNIK KOMPUTER DAN JARINGAN
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI UJUNG PANDANG
MAKASSAR
2022**

HALAMAN PENGESAHAN

Skripsi dengan judul **Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM dengan Menerapkan *Opposition-based Whale Optimization Algorithm* pada Klasifikasi SVM** oleh Asrul Said NIM 425 18 058 telah diterima dan disahkan sebagai salah satu syarat untuk memperoleh gelar Diploma IV (D-4/S1 Terapan) pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.

Makassar, 27 September 2022

Mengesahkan,

Pembimbing I,



Eddy Tungadi, S.T., M.T.
NIP. 19790823 201012 1 001

Pembimbing II,



Meylanie Olivya, S.T., M.T.
NIP. 19820503 201404 2 002

Mengetahui,

Koordinator Program Studi

Teknik Komputer dan Jaringan

Politeknik Negeri Ujung Pandang




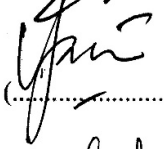
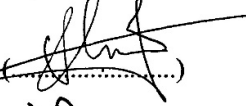
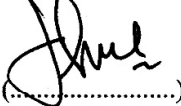

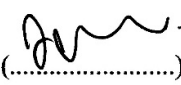
Eddy Tungadi, S.T., M.T.
NIP. 19790823 201012 1 001

HALAMAN PENERIMAAN

Pada hari ini, Selasa tanggal 27 September 2022, Tim Penguji Ujian Sidang Skripsi telah menerima dengan baik hasil skripsi oleh mahasiswa: **Asrul Said** nomor induk mahasiswa **42518058** dengan judul “**Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM dengan Menerapkan *Opposition-based Whale Optimization Algorithm* pada Klasifikasi SVM**”.

Makassar, 27 September 2022

Tim Penguji Ujian Sidang Skripsi:

- | | | |
|-------------------------------------------|------------|---------------------------------------------------------------------------------------|
| 1. Muh. Fajri Raharjo, S.T, M.T. | Ketua |  |
| 2. Muhammad Nur Yasir Utomo, S.ST, M.Eng. | Sekretaris |  |
| 3. Sulaeman, S.T., M.T. | Anggota |  |
| 4. Zawiyah Saharuna, S.T., M.Eng. | Anggota |  |
| 5. Eddy Tungadi, S.T., M.T. | Anggota |  |
| 6. Meylanie Olivya, S.T., M.T. | Anggota |  |

KATA PENGANTAR

Alhamdulillah puji syukur atas segala nikmat dan karunia tak terhingga yang diberikan oleh sang Maha Esa, Allah SWT, sehingga penulis mampu menyelesaikan skripsi ini dengan baik. Shalawat serta Salam senantiasa tercurah kepada Rasulullah SAW sebagai sebaik-baik panutan bagi seluruh umat manusia.

Sebagai salah satu syarat untuk menyelesaikan studi serta dalam rangka memperoleh gelar diploma IV (D-4/S1 Terapan) pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang, maka skripsi ini disusun dengan sebaik-baiknya. Penulis tentu menyadari bahwa keberhasilan skripsi ini tidak lepas dari bantuan berbagai pihak baik secara langsung maupun tidak langsung. Oleh karenanya, penulis menyampaikan apresiasi dengan menghaturkan terimakasih sebesar-besarnya kepada :

1. Orang Tua penulis yang sampai saat ini senantiasa memberikan dukungan kasih sayang, semangat, serta do'a terbaiknya sehingga penulis mampu menyelesaikan pendidikan jenjang D4/S1 Terapan dengan sangat baik.
2. Bapak Prof. Ir. Muhammad Anshar, M. Si., Ph.D. selaku Direktur Politeknik Negeri Ujung Pandang.
3. Bapak Ahmad Rizal Sultan, S.T., M.T., Ph.D. selaku ketua Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.
4. Bapak Eddy Tungadi, S.T., M.T. selaku koordinator Program Studi Teknik Komputer dan Jaringan dan selaku Pembimbing I yang dengan penuh ketulusan memberikan ilmu, motivasi, nasihat, bantuan serta kesedian waktu dalam membimbing penulis hingga terselesaikannya penelitian ini.
5. Ibu Meylanie Olivya, S.T., M.T. selaku dosen pembimbing II yang telah dengan sabar meluangkan waktu untuk memberikan ilmu, motivasi, nasihat, bantuan, dan bimbingan terbaik kepada penulis selama mengerjakan penelitian.
6. Seluruh Dosen dan Staf Jurusan Teknik Elektro, khususnya Program Studi D4 Teknik Komputer dan Jaringan yang telah memberikan bekal ilmu kepada penulis.

7. Teman-teman seperjuangan di Program Studi D-4 Teknik Komputer dan Jaringan angkatan 2018 yang mempunyai peranan besar dalam membantu menyusun skripsi ini dan mengajarkan banyak hal kepada penulis baik dari segi akademik maupun non akademik

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, sehingga penulis mengharap kritik dan saran yang membangun demi perbaikan di masa mendatang. Semoga tulisan ini bermanfaat.

Makassar, 27 September 2022

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
HALAMAN PENERIMAAN	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR TABEL.....	viii
DAFTAR GAMBAR	ix
SURAT PERNYATAAN.....	xi
RINGKASAN	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Ruang Lingkup Penelitian.....	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
BAB 2 TINJAUAN PUSTAKA	6
2.1 Penelitian Terkait	6
2.2 Analisis Sentimen	7
2.3 Ekstraksi Fitur	10
2.4 Seleksi Fitur	11
2.5 <i>Opposition-based Whale Optimization Algorithm (OBWOA)</i>	12
2.6 <i>Support Vector Machine (SVM)</i>	17
2.7 Evaluasi dan Validasi.....	23
BAB 3 METODOLOGI PENELITIAN.....	25
3.1 Tempat dan Waktu Penelitian	25
3.2 Alat dan Bahan	25

1. Alat.....	25
2. Bahan.....	26
3.3 Prosedur Penelitian.....	26
1. Pengumpulan dan Eksplorasi Data.....	26
2. <i>Pre-processing</i>	27
3. Ekstraksi Fitur	28
4. Optimasi Parameter SVM	29
5. Seleksi Fitur	33
6. Klasifikasi Sentimen	34
7. Validasi dan Evaluasi.....	35
8. Implementasi Model dan Visualisasi Hasil.....	35
 BAB 4 HASIL DAN PEMBAHASAN.....	 36
4.1 Pengumpulan dan Eksplorasi Data.....	36
4.2 Preprocessing	39
4.3 Ekstraksi Fitur	40
4.4 Hasil dan Analisis Optimasi Parameter SVM.....	41
4.5 Hasil dan Analisis Seleksi Fitur	43
4.6 Hasil dan Analisis Sentimen Masyarakat terhadap Kebijakan PPKM..	45
 BAB 5 KESIMPULAN DAN SARAN.....	 54
5.1 Kesimpulan	54
5.2 Saran.....	54
 DAFTAR PUSTAKA	 56
 LAMPIRAN	 60

DAFTAR TABEL

Tabel 2.1 Proses <i>filtering</i>	8
Tabel 2.2 Proses <i>case folding</i>	9
Tabel 2.3 Proses <i>slang word</i>	9
Tabel 2.4 Fungsi kernel SVM	19
Tabel 2.5 <i>Confusion Matrix</i>	23
Tabel 3.1 Skenario Pengujian untuk Mendapatkan Parameter Optimal SVM	30
Tabel 4.1 Hasil Tahapan Preprocessing	40
Tabel 4.2 Hasil Pengujian Optimasi Parameter SVM.....	41

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi <i>Bubble-net feeding</i> Paus Bungkuk	12
Gambar 2.2 <i>Pseudocode</i> algoritma WOA.....	16
Gambar 2.3 Skema algoritma OBWOA.....	17
Gambar 2.4 Ilustrasi algoritma SVM.....	18
Gambar 2.5 Ketika Nilai C Besar	21
Gambar 2.6 Ketika Nilai C Kecil.....	21
Gambar 2.7 Ketika Nilai <i>Gamma</i> Tinggi.....	22
Gambar 2.8 Ketika Nilai <i>Gamma</i> Rendah	22
Gambar 2.9 <i>K-fold cross validation</i>	24
Gambar 3.1 Tahapan <i>Pre-processing</i>	28
Gambar 3.2 Alur Proses Optimasi Parameter SVM Berbasis OBWOA.....	31
Gambar 3.3 Alur Proses Seleksi Fitur Menggunakan OBWOA.....	32
Gambar 3.4 Alur Klasifikasi Sentimen.....	34
Gambar 4.1 Hasil Pengambilan Data dari Twitter	36
Gambar 4.2 Karakteristik Data Hasil <i>Scrapping</i>	37
Gambar 4.3 Hasil Pengambilan Data dari Kaggle	37
Gambar 4.4 Karakteristik Data Latih	38
Gambar 4.5 Distribusi Jumlah setiap Label	38
Gambar 4.6 Hasil Ekstraksi Fitur.....	41
Gambar 4.7 Hasil Performa Pengujian Optimasi SVM	42
Gambar 4.8 Kurva Konvergensi Seleksi Fitur	43
Gambar 4.9 Perbandingan Hasil Performa Seleksi Fitur	44
Gambar 4.10 Perbandingan Waktu Komputasi.....	45

Gambar 4.11 Hasil Sentimen PPKM	46
Gambar 4.12 Contoh Tweet Sentimen Negatif, Netral, dan Positif	46
Gambar 4.13 Tren Sentimen Berdasarkan Waktu.....	47
Gambar 4.14 Tren Sentimen Bulan Januari 2021	47
Gambar 4.15 Contoh Tweet Sentimen Negatif di Bulan Januari	48
Gambar 4.16 Contoh Tweet Sentimen Netral di Bulan Januari.....	48
Gambar 4.17 Tren Sentimen Bulan Februari 2021	49
Gambar 4.18 Wordcloud Sentimen Positif Bulan Februari	49
Gambar 4.19 Tren Sentimen Bulan Juli – September 2021	50
Gambar 4.20 <i>Wordcloud</i> Sentimen Negatif Bulan Juli - September	50
Gambar 4.21 Contoh Tweet Sentimen Negatif di Bulan Juli.....	51
Gambar 4.22 Contoh Tweet Sentimen Positif di Bulan September.....	52
Gambar 4.23 Tren Sentimen Bulan Desember 2021	52
Gambar 4.24 Contoh Tweet Sentimen Negatif di Bulan Desember	53

SURAT PERNYATAAN

Saya bertanda tangan di bawah ini :

Nama : Asrul Said

NIM : 42518058

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam skripsi ini yang berjudul; “Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM dengan Menerapkan *Opposition-based Whale Optimization Algorithm* pada Klasifikasi SVM” merupakan gagasan dan hasil karya sendiri dengan arahan komisi pembimbing dan belum pernah diajukan dalam bentuk apa pun pada perguruan tinggi dan instansi mana pun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam skripsi ini.

Jika pernyataan saya tersebut di atas tidak benar, saya siap menanggung risiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, 27 September 2022



Asrul Said
NIM. 42518058

Analisis Sentimen Masyarakat Terhadap Kebijakan Ppkm Dengan Menerapkan *Opposition-Based Whale Optimization Algorithm* Pada Klasifikasi Svm

RINGKASAN

Salah satu kebijakan yang sedang ramai diperbincangkan masyarakat adalah penetapan PPKM. Dimana kebijakan tersebut menimbulkan pro dan kontra di masyarakat khususnya pada media sosial *twitter*. Salah satu pendekatan yang dapat digunakan untuk memahami respon dan persepsi masyarakat dari data *twitter* adalah analisis sentimen berbasis *machine learning*. Namun, pendekatan ini memiliki beberapa kekurangan pada data dengan dimensi fitur yang tinggi, fitur *redundant*, tidak relevan dan tumpang tindih sehingga berdampak pada proses komputasi karena memerlukan waktu dan memori yang lebih banyak. Oleh karena itu, penelitian ini mengusulkan seleksi fitur menggunakan *Opposition-based Whale Optimization* (OBWOA) pada analisis sentimen terhadap kebijakan PPKM berdasarkan *tweet* masyarakat di *twitter*. Pada penelitian ini juga digunakan *Support Vector Machine* (SVM) sebagai algoritma *classifier* Namun, performa SVM sepenuhnya bergantung pada pemilihan parameter yang disesuaikan dengan kondisi datanya. Sehingga, untuk mendapatkan performa klasifikasi yang baik maka terlebih dahulu dilakukan optimasi parameter SVM khususnya pada parameter C dan gamma. Berdasarkan hasil penelitian, diperoleh parameter SVM yang optimal C = 4.99522643 dan gamma = 1.4236435 dengan *accuracy*, *precision*, *recall*, dan *f-measure* masing-masing 75,20%, 79,73%, 71,65%, 70,88%. Selanjutnya, diperoleh fitur terbaik sebanyak 652 dengan *accuracy*, *precision*, *recall*, dan *f-measure* masing-masing 82,40%, 84,23%, 79,63%, dan 78,96%. Serta, metode seleksi fitur yang diusulkan juga mampu mengurangi waktu komputasi. Selain itu, berdasarkan hasil klasifikasi sentimen dapat diketahui bahwa mayoritas sentimen masyarakat terhadap kebijakan PPKM adalah Negatif.

Kata kunci: ppkm, analisis sentimen, seleksi fitur, optimasi parameter

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Corona Virus Disease 2019 (COVID-19) telah memberikan dampak yang serius pada berbagai sektor di Indonesia seperti kesehatan, bisnis, pariwisata, industri, hingga ekonomi. Berdasarkan data dari Satuan Tugas Penanganan COVID-19 pada 26 Juli 2021 sebanyak 3.194.733 terkonfirmasi positif, 2.549.692 pasien sembuh dan 84.766 meninggal dunia (Satuan Tugas Penanganan COVID-19, 2021). Berbagai kebijakan telah dilakukan pemerintah untuk menangani penyebaran COVID-19 seperti *physical distancing*, aturan wajib masker dan mencuci tangan secara berkala, vaksinasi, hingga Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM).

Salah satu kebijakan yang sedang ramai diperbincangkan masyarakat adalah penetapan PPKM. PPKM adalah kebijakan yang diambil pemerintah untuk membatasi mobilitas dan kegiatan masyarakat di luar rumah dengan tujuan agar dapat memutus rantai penyebaran virus *corona*. Selama PPKM, pemerintah menerapkan sejumlah kebijakan, dimana kebijakan tersebut menimbulkan pro dan kontra di masyarakat yang dapat dilihat melalui media sosial, khususnya *twitter* (Jayani, 2021). Menurut Ramadhanny (2021), sekitar 40 ribu *tweet* terkait respon masyarakat mulai dari dukungan, kritikan, saran, hingga sindiran terhadap pemerintah mengenai pelaksanaan PPKM.

Berdasarkan kondisi tersebut, pemerintah sebaiknya lebih berhati-hati dalam mengambil kebijakan, agar kebijakan tersebut mampu menjadi solusi atas kondisi yang dihadapi dengan tetap mempertimbangkan opini masyarakat. Oleh karena itu,

opini masyarakat khususnya di media sosial *twitter* dapat dijadikan sebagai salah satu sumber informasi yang efektif dalam membantu pemerintah mengambil kebijakan yang tepat. Namun, mengekstrak opini masyarakat di *twitter* bukanlah hal yang mudah. Ini dikarenakan kalimatnya yang tidak terstruktur dan sering kali ditulis dalam bahasa yang tidak baku (P. H. Prastyo et al., 2022). Oleh sebab itu, diperlukan suatu metode yang mampu mengekstrak opini tersebut.

Banyak metode analisis yang dapat digunakan untuk mengekstrak opini masyarakat berdasarkan informasi yang ada pada *twitter*, salah satunya adalah metode analisis sentimen (Laurensz & Sedyono, 2021). Analisis Sentimen adalah salah satu bidang studi dari *Natural Language Processing (NLP)* yang berfokus dalam mengidentifikasi suasana hati atau opini seseorang dalam sebuah teks. Prinsip dasar analisis sentimen adalah menentukan polaritas dari teks opini yang diberikan dan mengklasifikasikan opini tersebut menjadi sentimen positif, negatif atau netral (Jianqiang & Xiaolin, 2017).

Terdapat berbagai pendekatan analisis sentimen, salah satunya adalah pendekatan *machine learning*. Namun, menurut Suchetha et al. (2019) pendekatan ini memiliki beberapa kekurangan pada data dengan dimensi fitur yang tinggi, fitur *redundant*, tidak relevan dan tumpang tindih sehingga berdampak pada proses komputasi karena memerlukan waktu dan memori yang lebih banyak. Untuk mengatasi hal tersebut, maka diperlukan seleksi fitur. Seleksi fitur dapat mengurangi dimensi fitur, memilih fitur yang tepat, mengurangi waktu komputasi dan meningkatkan akurasi atau performa *classifier* (P. Prastyo et al., 2020). Metode yang sedang populer saat ini adalah metode *wrapper* berbasis metaheuristik karena memiliki performa yang baik dalam menyelesaikan masalah seleksi fitur. Salah satu

algoritma metaheuristik yang memiliki performa bagus, struktur sederhana dan mudah digunakan adalah *Whale Optimization Algorithm* (WOA) (Hassouneh et al., 2021). Namun, WOA memiliki kekurangan yakni mudah terjebak pada lokal optima. Untuk mengatasi kekurangan tersebut, Abd Elaziz & Oliva pada tahun 2018 mengajukan *Opposition-based Whale Optimization* (OBWOA) guna meningkatkan performa dari WOA.

Berdasarkan uraian di atas, maka pada penelitian ini mengusulkan seleksi fitur menggunakan *Opposition-based Whale Optimization* (OBWOA) pada analisis sentimen terhadap kebijakan PPKM berdasarkan *tweet* masyarakat di *twitter*.

Pada penelitian ini juga digunakan *Support Vector Machine* (SVM) sebagai algoritma *classifier* karena sangat populer dan memiliki performa yang baik terutama dalam menyelesaikan masalah analisis sentimen (Ruz et al., 2020). Namun, performa SVM sepenuhnya bergantung pada pemilihan parameter yang disesuaikan dengan kondisi datanya (Yin et al., 2019). Sehingga, untuk mendapatkan performa klasifikasi yang baik maka terlebih dahulu dilakukan optimasi parameter SVM khususnya pada parameter C dan gamma. Hasil penelitian ini diharapkan mampu menjadi alat bantu atau *Decision Support System* bagi pemerintah dalam membuat kebijakan.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang yang telah dipaparkan, maka rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana menerapkan OBWOA untuk mengoptimasi parameter pada algoritma SVM ?

2. Bagaimana menerapkan OBWOA pada proses seleksi fitur dari model hasil optimasi ?
3. Bagaimana mengetahui sentimen masyarakat terkait kebijakan PPKM menggunakan model yang telah dibangun?

1.3 Ruang Lingkup Penelitian

Adapun ruang lingkup pada penelitian ini adalah sebagai berikut:

1. Data yang digunakan untuk melatih model bersumber dari Kaggle Dataset (<https://www.kaggle.com/datasets/mochkholil/ppkm-sentiment>) yang diupload oleh Moch. Kholil dengan jumlah 300 baris data dan terdiri atas 3 label (Positif, Negatif, dan Netral).
2. Data yang digunakan untuk kebutuhan analisis adalah data *tweet* yang di-*scrapping* dari *twitter* dengan menggunakan kata kunci “PPKM”. Pengumpulan data dimulai pada tanggal 01 Januari 2021 hingga 31 Desember 2021.
3. Pengujian pada penelitian ini adalah dengan melakukan pengukuran performa algoritma yang terdiri dari *accuracy*, *precision*, *recall*, *f1-score* dan waktu komputasi.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Mengoptimasi parameter algoritma SVM pada lingkungan OBWOA.
2. Menerapkan OBWOA pada proses seleksi fitur dari model hasil optimasi.
3. Mengetahui sentimen masyarakat terkait kebijakan PPKM menggunakan model yang telah dibangun.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Memberikan kontribusi terhadap pengembangan dan penerapan sistem pengambilan keputusan (*decision support system*) bagi pemerintah dalam mengambil kebijakan.
2. Memberikan kontribusi pada penelitian terkait analisis sentimen dalam meningkatkan performa model machine learning dengan menerapkan algoritma OBWOA untuk mengoptimasi parameter SVM dan seleksi fitur.

BAB 2 TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Analisis sentimen merupakan salah satu bidang penelitian yang cukup populer dikembangkan sampai saat ini. Pada tahun 2020, Muhidin & Wibowo melakukan penelitian analisis sentimen terhadap sentimen masyarakat mengenai kebijakan *new normal* di masa pandemi. Penelitian tersebut membandingkan dua algoritma yaitu SVM dan *K-Nearest Neighbor* (KNN) untuk mengklasifikasikan *tweet* sentimen ke dalam kelas positif dan negatif. Hasil pengujian menunjukkan bahwa kinerja algoritma K-NN lebih baik dengan akurasi sebesar 72.96% dibanding algoritma SVM yang memiliki akurasi sebesar 71.19%. Pada penelitian ini, tidak dilakukan proses pembersihan terhadap kata-kata yang berupa singkatan ataupun kata tidak baku sehingga mempengaruhi akurasi dari algoritma klasifikasi (Jianqiang & Xiaolin, 2017). Di tahun yang sama, Prastyo et al. melakukan penelitian tentang analisis sentimen terkait respon masyarakat terhadap kebijakan pemerintah dalam menangani pandemi COVID-19. Peneliti pada penelitian tersebut, membandingkan dua algoritma yakni SVM dan Multinomial Naive Bayes (MNB). Hasilnya menunjukkan bahwa SVM memiliki tingkat akurasi yang lebih tinggi dibanding MNB dengan rata-rata akurasi sebesar 82%, presisi 82.24%, recall 82.01%, dan f-measure 81.84%. Namun, pada penelitian tersebut memiliki kekurangan yakni terlalu banyak fitur yang tidak relevan sehingga membuat algoritma klasifikasi membutuhkan komputasi yang cukup lama dalam memproses data (P. H. Prastyo, Sumi, et al., 2020).

Di tahun 2021, Gulati et al. membandingkan tujuh algoritma yaitu SVM, *Perceptron*, *Passive Aggressive Classifier*, *Logstic Regression*, *Multinomial NB*, *Bernoulli NB*, dan *AdaBoost Classifier* untuk mengklasifikasikan sentimen *tweet* terkait pandemi COVID-19. Dari hasil eksperimen, algoritma SVM berhasil mengungguli 6 algoritma lainnya yakni dengan akurasi sebesar 98.72% yang divalidasi sebanyak 10 kali menggunakan *K-fold cross validation*. Walaupun demikian, waktu komputasi dari algoritma SVM sangat lama jika dibandingkan dengan 6 algoritma lainnya yakni 174 detik hingga 869 detik sedangkan algoritma lainnya hanya membutuhkan waktu yang berkisar antara 31 detik hingga 577 detik.

Oleh karena itu, pada penelitian ini akan diterapkan seleksi fitur untuk mengurangi dimensi, menghapus fitur yang tidak relevan, memilih fitur yang tepat dan mengurangi waktu komputasi sehingga dapat meningkatkan performa dari algoritma klasifikasi. Pada penelitian ini juga, akan dilakukan pemilihan kernel dan optimasi parameter SVM. Hal ini dilakukan agar algoritma SVM memiliki konfigurasi parameter optimal sesuai dengan data yang akan digunakan (Yang & Shami, 2020).

2.2 Analisis Sentimen

Analisis Sentimen adalah salah satu bidang studi dari *Natural Language Processing (NLP)* yang berfokus dalam mengidentifikasi suasana hati atau opini seseorang dalam sebuah teks. Prinsip dasar analisis sentimen adalah menentukan polaritas dari teks opini yang diberikan dan mengklasifikasikan opini tersebut menjadi sentimen positif, negatif atau netral (Jianqiang & Xiaolin, 2017).

Terdapat tiga metode pada analisis sentimen, yaitu *machine learning approach*, *rule-based approach* dan *lexical based approach*. Akan tetapi dari ketiga metode

tersebut, analisis sentimen berbasis *machine learning* lebih populer karena memiliki performa yang lebih baik dibanding metode lainnya (Devika et al., 2016). Oleh karena itu, penelitian ini lebih difokuskan pada analisis sentimen berbasis *machine learning*.

Metode analisis sentimen berbasis *machine learning* terdiri dari beberapa tahapan seperti *pre-processing*, *feature extraction* (ekstraksi fitur), *feature selection* (seleksi fitur), tahapan klasifikasi, validasi hingga evaluasi. Adapun tahapan pada *pre-processing* terbagi menjadi beberapa proses, dimana penjelasan tiap proses tersebut adalah sebagai berikut:

1. *Filtering*

Filtering adalah proses untuk membersihkan atau menghilangkan karakter-karakter/*string* yang tidak penting dalam sebuah kalimat karena dapat mempengaruhi proses klasifikasi *machine learning*. Misalnya menghapus alamat situs (*http://*, *https://* atau *www.*), URL gambar (*pic.twitter.com*), tanda baca dan karakter lainnya.

Tabel 2.1 Proses *filtering*

Sebelum <i>filtering</i>	Sesudah <i>filtering</i>
Hasil PPKM yang nggak jelas... Tutup jam 8, buka lagi jam 10 sampai pagi... berkerumun! Aneh kalau ini pun tidak di-pidanakan dengan pelanggaran prokes!! https://t.co/jXCWlxtyKZ	Hasil PPKM yang nggak jelas Tutup jam buka lagi jam sampai pagi berkerumun Aneh kalau ini pun tidak dipidanakan dengan pelanggaran prokes

2. *Case Folding*

Case folding adalah proses menyeragamkan setiap huruf dalam kalimat. Misalnya mengubah huruf kapital menjadi huruf kecil.

Tabel 2.2 Proses *case folding*

Sebelum <i>case folding</i>	Sesudah <i>case folding</i>
Hasil PPKM yang nggak jelas Tutup jam buka lagi jam sampai pagi berkerumun Aneh kalau ini pun tidak dipidanakan dengan pelanggaran prokes	hasil ppkm yang nggak jelas tutup jam buka lagi jam sampai pagi berkerumun aneh kalau ini pun tidak dipidanakan dengan pelanggaran prokes

3. *Slang Word*

Slang word adalah proses mengubah kata yang tidak baku menjadi kata baku.

Tabel 2.3 Proses *slang word*

Sebelum <i>slang word</i>	Sesudah <i>slang word</i>
hasil ppkm yang nggak jelas tutup jam buka lagi jam sampai pagi berkerumun aneh kalau ini pun tidak dipidanakan dengan pelanggaran prokes	hasil ppkm yang tidak jelas tutup jam buka lagi jam sampai pagi berkerumun aneh kalau ini pun tidak dipidanakan dengan pelanggaran prokes

4. *Stopword Removal*

Stopword removal merupakan proses menghapus kata-kata yang tidak penting dengan memeriksa setiap kata berdasarkan *stopwordlist*. Contoh kata *stopword* dalam bahasa Indonesia seperti: “yang”, “di”, “pun”, “dari” dan lain-lain.

Tabel 2.4 Proses *stopword removal*

Sebelum <i>stopword removal</i>	Sesudah <i>stopword removal</i>
“hasil” “ppkm” “yang” “tidak” “jelas” “tutup” “jam” “buka” “lagi” “jam” “sampai” “pagi” “kerumun” “aneh” “kalau” “ini” “pun” “tidak” “pidana” “dengan” “langgar” “prokes”	“hasil” “ppkm” “tidak” “jelas” “tutup” “jam” “buka” “lagi” “jam” “sampai” “pagi” “kerumun” “aneh” “kalau” “tidak” “pidana” “dengan” “langgar” “prokes”

5. *Stemming*

Stemming merupakan proses menghilangkan imbuhan meliputi awalan kata (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) yang terdapat pada kata sehingga kata tersebut menjadi kata dasar. Contoh algoritma *stemming* untuk bahasa Indonesia adalah Algoritma Nazief & Adriani.

Tabel 2.5 Proses *stemming*

Sebelum <i>stemming</i>	Sesudah <i>stemming</i>
hasil ppkm yang tidak jelas tutup jam buka lagi jam sampai pagi berkerumun aneh kalau ini pun tidak dipidanakan dengan pelanggaran prokes	hasil ppkm yang tidak jelas tutup jam buka lagi jam sampai pagi kerumun aneh kalau ini pun tidak pidana dengan langgar prokes

6. *Tokenization*

Tokenization adalah proses memecah atau memisahkan kalimat menjadi beberapa kata (*token*).

Tabel 2.6 Proses *tokenization*

Sebelum <i>tokenization</i>	Sesudah <i>tokenization</i>
hasil ppkm yang tidak jelas tutup jam buka lagi jam sampai pagi kerumun aneh kalau ini pun tidak pidana dengan langgar prokes	“hasil” “ppkm” “yang” “tidak” “jelas” “tutup” “jam” “buka” “lagi” “jam” “sampai” “pagi” “kerumun” “aneh” “kalau” “ini” “pun” “tidak” “pidana” “dengan” “langgar” “prokes”

2.3 Ekstraksi Fitur

Ekstraksi fitur merupakan proses untuk mengubah kata menjadi vektor sehingga dapat diproses oleh *machine learning*. Salah satu metode ekstraksi fitur yang umum digunakan adalah *Term Frequency-Inverse Document Frequency* (TF-IDF).

Prinsip kerja dari *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah dengan menghitung bobot setiap kata yang muncul pada dokumen (Sutoyo & Almaarif, 2020). TF menghitung seberapa sering muncul kata tersebut dalam dokumen/*tweet*. Sedangkan IDF menghitung seberapa penting kata tersebut dengan cara memberi nilai rendah pada kata yang sering muncul. Nilai TF dapat dihitung menggunakan pada Persamaan (2-1)

$$TF_{(t,d)} = \frac{f_{t,d}}{\Sigma t,d} \quad (2-1)$$

Dimana:

$f_{t,d}$ = frekuensi tiap kata(t) muncul dalam dokumen (d)

$\Sigma t, d$ = total seluruh kata (t) yang ada dalam dokumen (d)

Untuk nilai IDF dapat dihitung menggunakan Persamaan (2-2).

$$IDF_{(t)} = \log \frac{|D|}{df(t)+1} \quad (2-2)$$

Dimana:

$|D|$ = jumlah dokumen yang ada dalam koleksi

$df(t)$ = jumlah dokumen tempat munculnya kata (t)

Untuk nilai TF-IDF dihitung menggunakan Persamaan (2-3).

$$TF - IDF = TF_{(t,d)} \cdot IDF_{(t)} \quad (2-3)$$

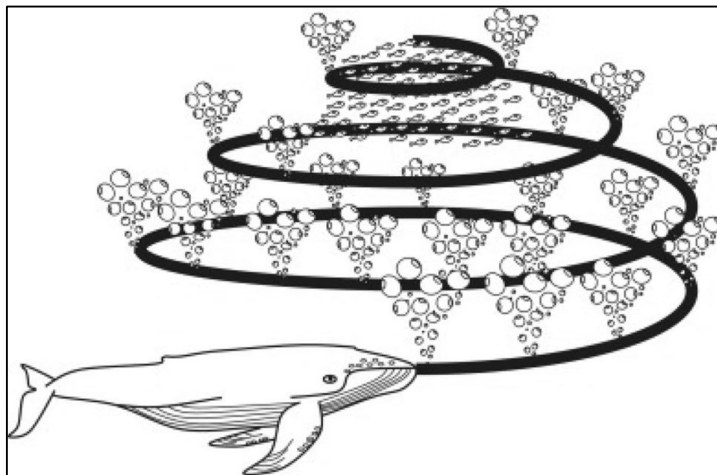
2.4 Seleksi Fitur

Seleksi fitur merupakan proses untuk memilih fitur terbaik yang relevan terhadap data. Seleksi fitur dapat mengurangi dimensi fitur, memilih fitur yang tepat, mengurangi waktu komputasi dan meningkatkan akurasi atau performa *classifier* (Suchetha et al., 2019). Seleksi fitur digunakan untuk mengatasi masalah

pada algoritma *machine learning* yaitu dimensi fitur yang tinggi, fitur *redundant*, tidak relevan dan tumpang tindih sehingga berdampak pada proses komputasi karena memerlukan waktu dan memori yang lebih banyak.

Penelitian terkait metode seleksi fitur saat ini telah berkembang sangat pesat. Dalam beberapa tahun terakhir, algoritma metaheuristik dianggap sebagai metode yang andal dan paling efisien terutama dalam memecahkan masalah dimensi fitur yang tinggi (Too & Mirjalili, 2021). Beberapa contoh algoritma metaheuristik yaitu *Particle Swarm Optimization* (PSO), *Genetic Algorithm* (GA), *Grey Wolf Optimizer* (GWO), dan *Flower Pollination Algorithm* (FPA). Lalu pada tahun 2016, Mirjalili & Lewis mengusulkan algoritma metaheuristik dengan nama *Whale Optimization Algorithm* (WOA).

2.5 *Opposition-based Whale Optimization Algorithm* (OBWOA)



Gambar 2.1 Ilustrasi *Bubble-net feeding* Paus Bungkuk

Sumber: (Mirjalili & Lewis, 2016)

Whale Optimization Algorithm (WOA) merupakan salah satu algoritma metaheuristik yang diusulkan oleh Mirjalili & Lewis pada tahun 2016. Ide tersebut terinspirasi dari kebiasaan paus bungkuk dalam mencari dan menyerang mangsa,

seperti yang ditunjukkan pada Gambar 2.1. Saat berburu, paus bungkuk berenang di bawah mangsanya sembari mengeluarkan gelembung. Gelembung ini berguna sebagai penghalang agar mangsanya tidak dapat kabur. Kemudian paus bungkuk berenang ke atas membentuk angka ‘9’ atau spiral untuk mempersempit ruang gerak dan secara bersama-sama membuka mulut dan memakan mangsanya.

Algoritma WOA terdiri atas tiga fase, yakni *encircling prey*, *spiral bubble-net feeding maneuver*, dan *search for prey* (Mirjalili & Lewis, 2016). Ketiganya dijelaskan sebagai berikut :

1. *Encircling prey*

Pada fase ini paus bungkuk mengenali lokasi mangsa dan mengelilinginya. Karena posisi optimal dalam ruang pencarian tidak diketahui skala prioritasnya, maka WOA mengasumsikan bahwa solusi terbaik saat ini adalah target/mangsa yang dikelilingi atau dengan kata lain posisi saat ini adalah yang mendekati optimal. Setelah *search agent* (paus pencari) terbaik ditentukan, maka paus lainnya akan memperbarui posisi mereka mendekati posisi *search agent* terbaik. Perilaku ini dapat diformulasikan melalui persamaan berikut:

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \quad (2-4)$$

$$\vec{X}(t + 1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \quad (2-5)$$

Dimana t menunjukkan iterasi saat ini, \vec{A} dan \vec{C} merupakan koefisien vektor, \vec{X}^* adalah vektor posisi dari solusi terbaik saat ini, \vec{X} merupakan vektor posisi yang akan diperbarui. \vec{X}^* harus diperbarui tiap iterasi jika

terdapat solusi yang lebih baik dari iterasi sebelumnya. Vektor \vec{A} dan \vec{C} dapat dihitung menggunakan persamaan berikut:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (2-6)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (2-7)$$

$$\vec{a} = \vec{a} - t \frac{\vec{a}}{t_{max}} \quad (2-8)$$

Dimana \vec{a} berkurang secara linier dari 2 ke 0 selama iterasi, baik pada fase eksplorasi maupun fase eksploitasi. \vec{r} merupakan vektor *random* pada [0,1].

2. *Bubble-net attacking method*

Bubble-net attacking atau fase eksploitasi terbagi dalam dua pendekatan sebagai berikut:

a. *Shrinking encircling mechanism*

Mekanisme *shrinking encircling* berarti paus mendekat dan mengelilingi mangsanya. Mekanisme ini diperoleh dari pengurangan nilai \vec{a} pada persamaan (2-6). Perlu diperhatikan bahwa nilai \vec{A} juga berkurang seiring dengan berkurangnya nilai \vec{a} . Dengan kata lain, \vec{A} adalah nilai acak pada interval $[-a, a]$ dimana a berkurang selama iterasi.

b. *Spiral updating position*

Mekanisme *spiral updating position* menunjukkan gaya berenang paus bungkuk yang berbentuk spiral. Pendekatan ini pertama kali menghitung jarak antara lokasi paus di posisi (X, Y) ke posisi dimana

mangsa berada (X^*, Y^*) . Adapun persamaan spiral yang terbentuk antara posisi paus dengan mangsa adalah sebagai berikut:

$$\vec{X}(t + 1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (2-9)$$

Dimana $\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|$ merupakan jarak antara posisi paus dengan mangsa (solusi terbaik), b adalah konstanta yang digunakan untuk menentukan bentuk dari logaritma spiral, l yaitu angka acak dalam rentang $[-1,1]$.

Paus bungkuk dapat berenang secara bersamaan di sekitar mangsanya dalam bentuk lintasan spiral dan melingkar untuk mempersempit ruang gerak mangsanya. Perilaku ini dapat dimodelkan melalui persamaan berikut:

$$\vec{X}(t + 1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}(t) & \text{if } p \geq 0.5 \end{cases} \quad (2-10)$$

Dimana $p \in [0,1]$ merupakan peluang paus memilih mekanisme renang (*shrinking encircling mechanism* atau model spiral).

3. *Search for prey* (fase eksplorasi)

Paus bungkuk mencari mangsa secara *random* dengan memperhatikan posisi satu sama lain. Model matematisnya ditunjukkan melalui persamaan berikut:

$$\vec{D} = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}| \quad (2-11)$$

$$\vec{X}(t + 1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \quad (2-12)$$

Dimana \vec{X}_{rand} adalah vektor posisi acak (*random whale*) yang dipilih dari populasi saat ini.

Algoritma Whale Optimization diawali dari proses iterasi dengan memilih solusi acak. Dalam tiap iterasi, jika $|A| > 1$, maka *search agent* akan dipilih secara acak. Sedangkan jika $|A| < 1$, maka posisi terbaik saat ini akan dipilih untuk memperbarui posisi dari *search agent*.

```

Initialize the whales population  $X_i(i = 1, 2, \dots, n)$ 
Calculate the fitness of each search agent
 $X^*$  = the best search agent
while ( $t <$  maximum number of iterations)
  for each search agent
    Update  $a, A, C, l$  and  $p$ 
    if1 ( $p < 0.5$ )
      if2 ( $|A| < 1$ )
        Update the position of the current search agent by the Eq. (2 – 4)
      else if2 ( $|A| \geq 1$ )
        Select a random search agent ( $X_{rand}$ )
        Update the position of the current search agent by the Eq. (2 – 11)
      end if2
    else if1 ( $p \geq 0.5$ )
      Update the position of the current search by the Eq. (2 – 8)
    end if1
  end for
  Check if any search agent goes beyond the search space and amend it
  Calculate the fitness of each search agent
  Update  $X^*$  if there is a better solution
   $t = t + 1$ 
end while
return  $X^*$ 

```

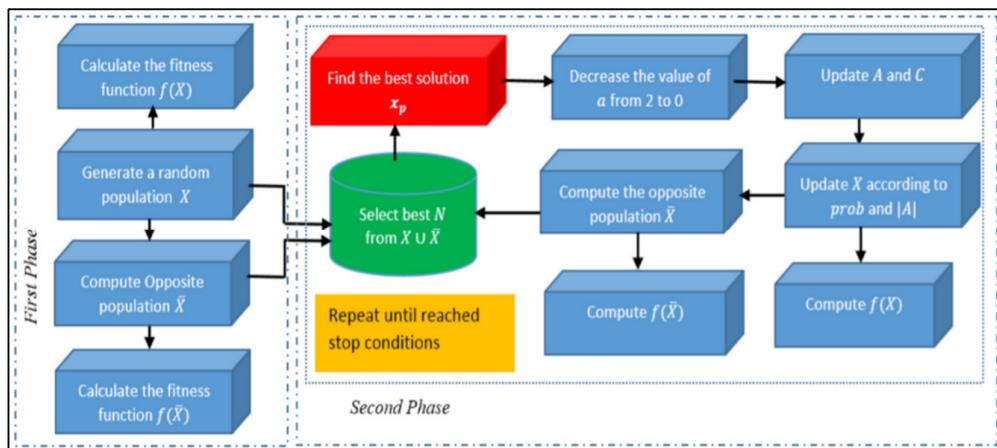
Gambar 2.2 Pseudocode algoritma WOA

Sumber: (Mirjalili & Lewis, 2016)

Dengan bergantung pada nilai p , WOA dapat memilih metode pergerakan antara berenang secara spiral atau berenang secara melingkar/*encircle* hingga algoritma

memenuhi kriteria dan proses iterasi selesai. Secara umum, *pseudocode* algoritma WOA ditunjukkan pada Gambar 2.2.

Pada tahun 2018, Abd Elaziz & Oliva melakukan pengembangan terhadap algoritma WOA. Pada penelitian tersebut, *opposition based learning* (OBL) digunakan untuk meningkatkan eksplorasi ruang pencarian algoritma WOA. OBL berfungsi untuk menghasilkan posisi berlawanan dari kandidat solusi terbaik saat ini pada waktu yang sama. Oleh karena itu, eksplorasi ruang pencarian WOA meningkat dan akan ada lebih banyak kemungkinan untuk menemukan solusi *global optima* yang ditunjukkan dengan adanya konvergensi pada proses optimasi (Abd Elaziz & Oliva, 2018). Skema algoritma OBWOA dapat dilihat ada gambar berikut:



Gambar 2.3 Skema algoritma OBWOA

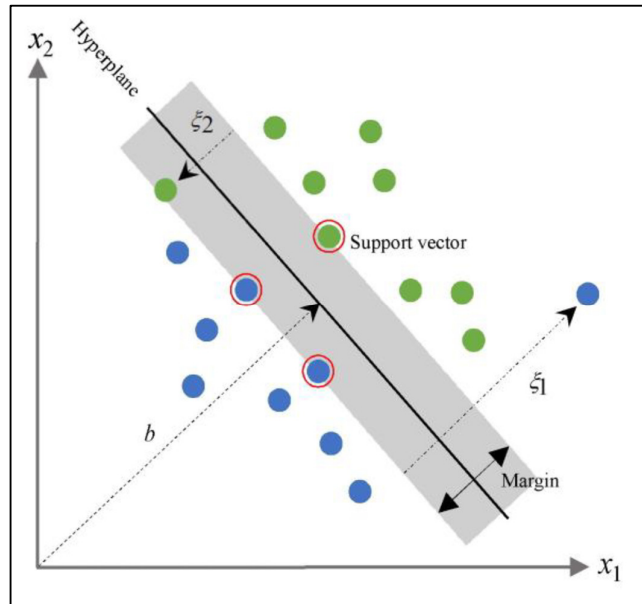
Sumber: (Abd Elaziz & Oliva, 2018)

2.6 Support Vector Machine (SVM)

Klasifikasi sentimen pada penelitian ini menggunakan metode *machine learning* dengan algoritma *Support Vector Machine* (SVM). SVM merupakan algoritma yang banyak digunakan untuk menyelesaikan masalah prediksi, baik itu

klasifikasi maupun regresi. SVM tergolong ke dalam *supervised learning* yaitu membutuhkan label pada saat proses *training*.

Ide utama dari algoritma ini adalah menemukan *hyperplane* terbaik untuk memisahkan dua buah kelas. *Hyperplane* adalah sebuah fungsi yang dapat digunakan sebagai pemisah antar kelas. *Hyperplane* terbaik dapat diperoleh dengan memaksimalkan jarak pemisah dengan titik terdekat (*support vectors*). Seiring dengan perkembangannya, SVM tidak hanya mampu memisahkan dua buah kelas tetapi kini juga dapat memisahkan kelas yang lebih banyak (Delimayanti et al., 2021).



Gambar 2.4 Ilustrasi algoritma SVM

Sumber: (Nanda et al., 2018)

Pada algoritma SVM, terdapat beberapa fungsi kernel yang sangat penting dalam menyelesaikan permasalahan non-linear. Adapun beberapa fungsi kernel dari SVM dapat dilihat pada Tabel 2.4, dimana C : cost, γ : gamma, r : coefficient, d : degree.

Tabel 2.4 Fungsi kernel SVM

No.	Fungsi Kernel	Rumus	Parameter
1	Linear	$K(x_n, x_i) = (x_n, x_i)$	C dan γ
2	RBF	$K(x_n, x_i) = \exp(-\gamma \ x_n - x_i\ ^2 + C)$	C dan γ
3	Sigmoid	$K(x_n, x_i) = \tanh(\gamma(x_n, x_i) + r)$	$C, \gamma, \text{ dan } r$
4	Polynomial	$K(x_n, x_i) = (\gamma(x_n, x_i) + r)^d$	$C, \gamma, r, \text{ dan } d$

Sumber: (Nanda et al., 2018)

Pemilihan kernel dan parameter merupakan hal yang sangat penting untuk meningkatkan performa algoritma SVM (P. H. Prastyo, Ardiyanto, et al., 2020). Ini dikarenakan konfigurasi parameter SVM berbeda-beda untuk setiap permasalahan yang akan diselesaikan. Berikut beberapa fungsi kernel pada algoritma SVM (Sitorus, 2020).

1. Kernel Linear

Kernel linear adalah fungsi kernel yang paling sederhana. Kernel linear umumnya digunakan pada data dengan dimensi yang lebih rendah. Dimensi yang lebih rendah dari kumpulan data berarti memiliki lebih sedikit fitur untuk diklasifikasi.

2. Kernel RBF

Kernel RBF atau juga disebut kernel Gaussian adalah konsep kernel yang paling banyak digunakan untuk memecahkan masalah klasifikasi data yang tidak dapat dipisahkan secara linear. Kernel ini dikenal memiliki performa yang baik dengan parameter tertentu, dan hasil dari pelatihan memiliki nilai error yang kecil dibandingkan dengan kernel lainnya.

3. Kernel Sigmoid

Kernel sigmoid merupakan pengembangan dari jaringan syaraf tiruan. Kernel sigmoid telah diusulkan secara teoritis untuk SVM, namun hingga saat ini belum banyak digunakan dalam praktiknya.

4. Kernel Polynomial

Kernel polynomial adalah bentuk yang lebih umum dari kernel linear. Kernel polynomial cocok untuk memecahkan masalah klasifikasi pada dataset pelatihan yang dinormalisasi. Kernel polynomial memiliki parameter *degree* (d) yang berfungsi untuk mencari nilai optimal pada setiap dataset. Parameter d adalah derajat dari fungsi kernel polynomial dengan nilai default $d = 2$. Semakin besar nilai d maka akurasi sistem yang dihasilkan akan fluktuatif dan kurang stabil. Hal ini terjadi karena semakin tinggi nilai parameter d maka semakin melengkung garis hyperplane yang dihasilkan.

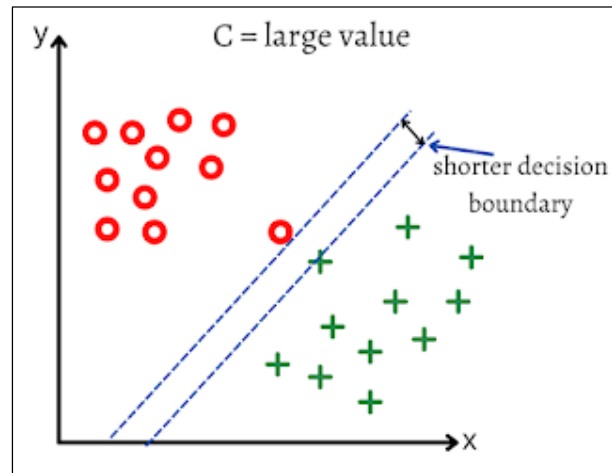
Terdapat dua parameter penting dalam menentukan garis *hyperplane* terbaik pada SVM (Kumar, 2020; Trivusi, 2022), antara lain sebagai berikut:

1. *Cost* (C)

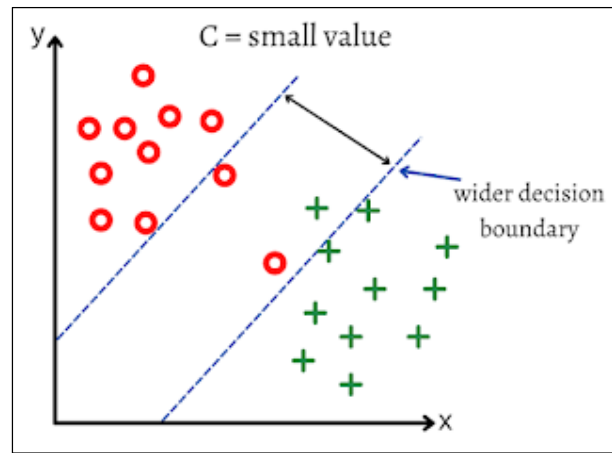
Parameter *Cost* atau biasa disebut sebagai C adalah parameter yang bekerja sebagai pengontrol SVM untuk menghindari kesalahan klasifikasi di setiap sampel pada saat *training*. Ketika nilai C terlalu besar, algoritma mencoba untuk mengurangi kesalahan klasifikasi sebanyak mungkin. Sederhananya jika C terlalu besar akan menyebabkan *decision boundary* menjadi sangat kecil. Ilustrasi ketika nilai C Besar dapat dilihat pada Gambar 2.5.

Sedangkan ketika nilai C terlalu kecil, kesalahan klasifikasi akan terjadi karena *decision boundary* yang lebih luas. *Decision boundary* yang lebih luas

digeneralisasikan dengan baik pada data pelatihan dan pengujian tetapi dapat mengakibatkan kesalahan klasifikasi pada beberapa sampel data. Ilustrasi ketika nilai C Kecil dapat dilihat pada Gambar 2.6.



Gambar 2.5 Ketika Nilai C Besar

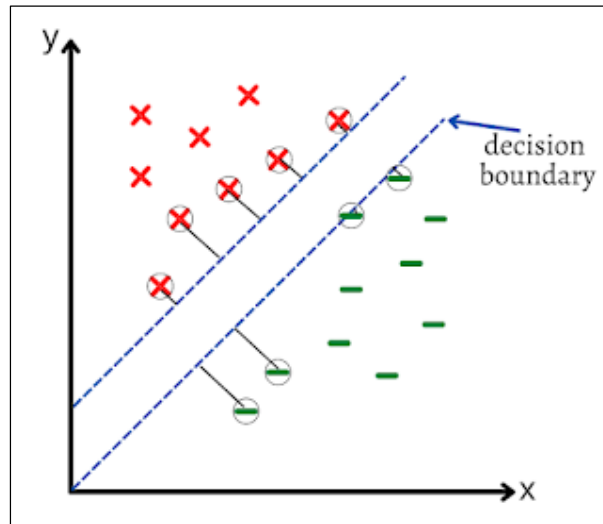


Gambar 2.6 Ketika Nilai C Kecil

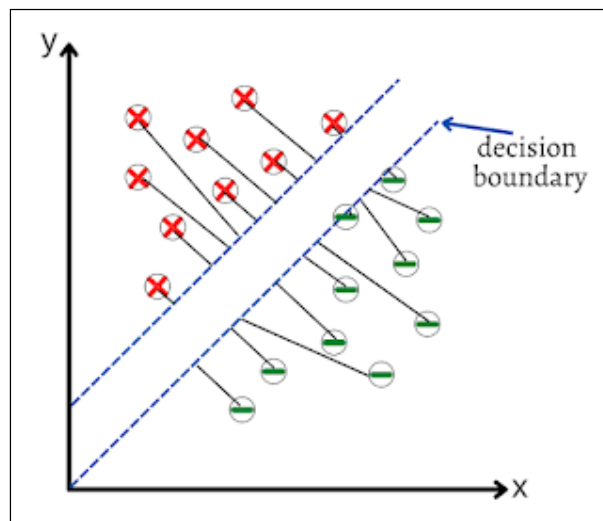
2. *Gamma*

Parameter *gamma* menentukan seberapa jauh pengaruh dari satu sampel dataset pelatihan. Nilai rendah berarti “jauh”, dan nilai yang tinggi berarti “dekat”. Dengan nilai *gamma* tinggi, *decision boundary* hanya akan tergantung

pada titik-titik yang sangat dekat dengannya, seperti yang terlihat pada Gambar 2.7. Sedangkan nilai γ yang rendah menunjukkan bahwa bahkan titik yang jauh pun dipertimbangkan ketika kita hendak memutuskan di mana *decision boundary* seharusnya berada, seperti yang terlihat pada Gambar 2.8.



Gambar 2.7 Ketika Nilai γ Tinggi



Gambar 2.8 Ketika Nilai γ Rendah

2.7 Evaluasi dan Validasi

Penelitian ini menggunakan *confusion matrix* untuk proses evaluasi. *Confusion Matrix* merupakan salah satu teknik evaluasi yang digunakan untuk mengukur kinerja dari algoritma klasifikasi. Metode perhitungan *confusion matrix* adalah dengan perbandingan hasil klasifikasi algoritma *machine learning* (prediksi) dengan hasil klasifikasi yang seharusnya (aktual). Terdapat empat indikator pengukuran pada *confusion matrix* yaitu *accuracy*, *recall*, *precision* dan *F1 score* (P. H. Prastyo, Sumi, et al., 2020).

Tabel 2.5 *Confusion Matrix*

		Aktual	
		1	0
Prediksi	1	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
	0	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Sumber: (P. H. Prastyo, Sumi, et al., 2020)

Keempat indikator pengukuran *confusion matrix* dapat dihitung menggunakan persamaan berikut:

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)} \quad (2-13)$$

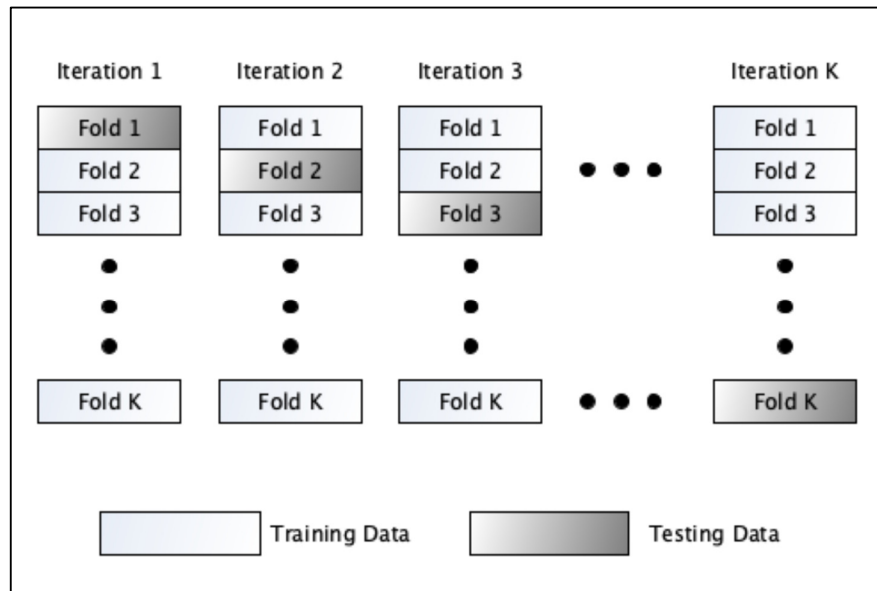
$$Recall = \frac{TP}{(TP+FN)} \quad (2-14)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (2-15)$$

$$F1\ Score = 2 \times \frac{precision \times recall}{precision+recall} \quad (2-16)$$

Setelah itu, algoritma klasifikasi kemudian di validasi menggunakan *K-fold cross-validation*. Metode ini digunakan untuk memisahkan data menjadi *K* bagian. Tiap iterasi, data akan dibagi menjadi data *training* dan *testing*. Hal ini dilakukan sebanyak nilai dari *K*, sehingga seluruh data mendapatkan kesempatan menjadi data

training dan data *testing*. Skema dari *K-fold cross validation* dapat dilihat pada Gambar 2.9.



Gambar 2.9 *K-fold cross validation*
Sumber: (P. H. Prastyo, Sumi, et al., 2020)

BAB 3 METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Laboratorium Sistem Informasi Gedung Administrasi Lantai 3 Kampus 1 Politeknik Negeri Ujung Pandang, Jl. Perintis Kemerdekaan KM 10 Makassar. Dimulai pada bulan Februari 2022 sampai dengan Juli 2022.

3.2 Alat dan Bahan

1. Alat

Kebutuhan sistem yang akan digunakan pada penelitian ini terbagi menjadi dua yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*).

a. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan yaitu 1 unit laptop *Lenovo Ideapad Slim 3*, processor AMD *Ryzen 7-4700U* 2.0GHz dan *memory* 8GB DDR4.

b. Perangkat Lunak (*Software*)

- Python 3.8.6 merupakan bahasa pemrograman yang digunakan untuk mengambil dan mengolah data serta untuk membangun sistem analisis sentimen.
- Jupyter Notebook 6.1.4 digunakan untuk menjalankan sistem yang berbasis pada bahasa pemrograman *python*.
- *Google Colaboratory* digunakan untuk menjalankan algoritma analisis sentimen dengan memanfaatkan *library* seperti *Scikit-learn*, *NLTK*, *Numpy*, *Pandas* dll.

- Snsrape versi 0.3.5 adalah library *python* yang digunakan untuk melakukan *scrapping* data dari *twitter* (JustAnotherArchivist, 2018).

2. Bahan

Bahan yang digunakan dalam penelitian ini ada dua yaitu data *tweet* opini masyarakat terkait kebijakan PPKM yang diperoleh melalui media sosial *twitter* dan data latih yang diperoleh dari Kaggle Dataset.

3.3 Prosedur Penelitian

Prosedur penelitian diperlukan agar penelitian dapat berjalan secara terstruktur, sehingga hasil yang akan diperoleh sesuai dengan tujuan penelitian. Adapun prosedur pada penelitian ini adalah sebagai berikut:

1. Pengumpulan dan Eksplorasi Data

Pada tahap pengumpulan data, peneliti menggunakan dua jenis sumber data, yaitu data primer dan data sekunder. Data primer merupakan data yang dikumpulkan sendiri oleh peneliti dan akan digunakan untuk kebutuhan analisis pada penelitian ini. Pengumpulan data dilakukan menggunakan metode web *scrapping* pada situs *twitter.com*. Data yang dikumpulkan merupakan *tweet* opini masyarakat terkait kebijakan PPKM dengan periode pengumpulan data mulai 01 Januari 2021 hingga 31 Desember 2021. Proses ini dilakukan dengan menggunakan library *snsrape* dan kata kunci yang digunakan yaitu “PPKM”. Data yang berhasil dikumpulkan kemudian diproses untuk menghilangkan data yang duplikat dan tidak relevan dengan topik PPKM.

Data sekunder merupakan data yang telah dikumpulkan dan dianotasi dari penelitian rujukan. Tujuan penggunaan data sekunder ini adalah sebagai data latih pada tahapan pembuatan model.

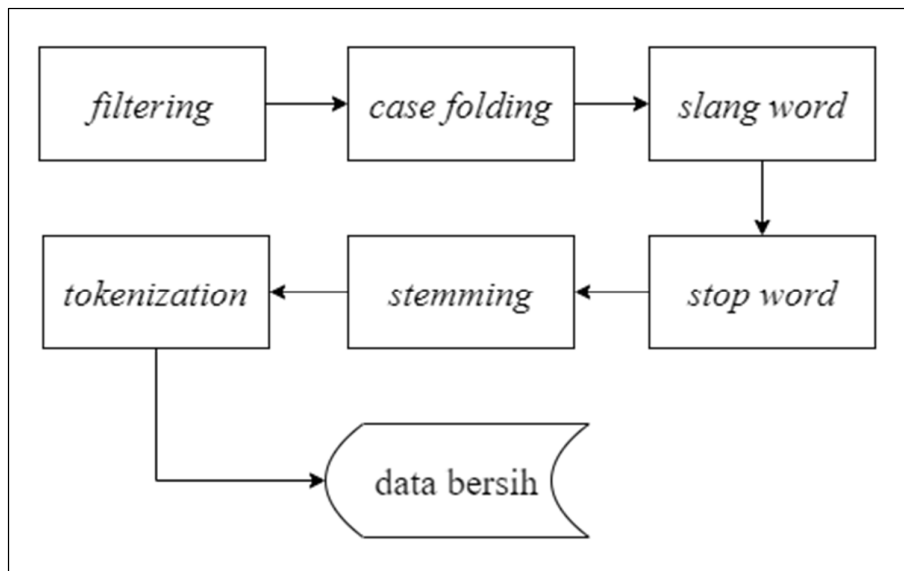
Selanjutnya, data yang telah dikumpulkan akan dieksplorasi untuk memahami dan mengetahui karakteristik dari setiap data sebelum dilakukan tahapan preprocessing.

2. *Pre-processing*

Data yang terkumpul merupakan data yang tidak terstruktur dan tidak baku. Oleh karena itu, tahap ini dilakukan agar data menjadi bersih dan seragam sehingga dapat diolah dengan baik oleh algoritma *machine learning*. Adapun alur dari tahapan *pre-processing* dapat dilihat pada Gambar 3.1.

- a. *Filtering* adalah proses untuk membersihkan data dari karakter/string yang tidak diperlukan seperti *@username*, alamat situs dan gambar (*http://*, *https://*, *www*, atau *pic.twitter.com*), tanda baca (koma, titik, tanda tanya, tanda seru), angka, dan karakter lainnya.
- b. *Case Folding* adalah proses menyeragamkan setiap kata menjadi huruf kecil.
- c. *Slang Word* adalah proses mengubah kata tidak baku menjadi kata baku.
- d. *Stopword Removal* yaitu proses untuk menghapus kata-kata yang tidak penting dengan memeriksa setiap kata berdasarkan *stopwordlist*. Adapun *stopwordlist* yang digunakan pada penelitian ini bersumber dari <https://www.kaggle.com/datasets/mochkholil/ppkm-sentiment?select=stopwordv1.txt> .

- e. *Stemming* adalah proses menghilangkan imbuhan kata sehingga kata tersebut menjadi kata dasar. Pada tahapan ini menggunakan *library Sastrawi Python*.
- f. *Tokenization* adalah proses untuk memisahkan setiap kata yang menyusun suatu dokumen/*tweet*. Tahap ini akan menggunakan *library NLTK*.



Gambar 3.1 Tahapan *Pre-processing*

3. Ekstraksi Fitur

Data *tweet* yang telah bersih, selanjutnya akan diubah menjadi bentuk vektor sehingga dapat diproses oleh *machine learning*. Pada tahap ini akan menggunakan *Term Frequency – Inverse Document Frequency (TF-IDF)* sebagai ekstraksi fitur. Prinsip kerja dari *TF-IDF* adalah menghitung bobot setiap kata berdasarkan frekuensi kemunculan kata tersebut dalam sebuah dokumen/*tweet*. Adapun library yang akan digunakan pada tahapan ini adalah *Tfidfvectorizer* dari *sklearn*.

4. Optimasi Parameter SVM

Tahapan optimasi SVM dilakukan dengan menerapkan algoritma OBWOA untuk mencari parameter SVM yang cocok dengan data yang dipakai pada penelitian ini. Pada proses ini, OBWOA berinteraksi langsung dengan algoritma SVM untuk mengevaluasi setiap individu menggunakan nilai *fitness*. Nilai *fitness* dihitung berdasarkan nilai error yang didapatkan dari hasil pengurangan akurasi klasifikasi.

$$Fitness = ER$$

Dimana *ER* adalah nilai error klasifikasi yang didapatkan dari $1 - P$. P adalah ukuran performa algoritma *machine learning*, dalam hal ini rata-rata akurasi. Semakin kecil nilai *Fitness* maka semakin baik pula performa SVM. Adapun proses kerja optimasi parameter berbasis OBWOA ini dapat dilihat pada Gambar 3.2.

Pada Gambar 3.2 proses optimasi diawali dengan menginisialisasi parameter OBWOA seperti jumlah populasi (N), jumlah iterasi maksimum (max_iter), nilai batas atas (ub) dan nilai batas bawah (lb) serta dimensi atau jumlah parameter yang akan dioptimasi dalam hal ini nilai C dan γ . Selanjutnya, nilai posisi awal X (kandidat nilai C dan γ) dinisialisasi dan nilai X_0 dihitung berdasarkan persamaan 3-1

$$\vec{x} = u_i + l_i - x_i, i = 1, 2, \dots, n \quad (3-1)$$

Setelah itu, dilakukan klasifikasi pada dataset yang divalidasi dengan menggunakan *5-fold cross validation*. Proporsi pembagian dataset adalah 80% untuk training dan 20% untuk testing. Algoritma SVM digunakan untuk mencari nilai rata-rata akurasi sebagai masukan pada fungsi *fitness*. Kemudian

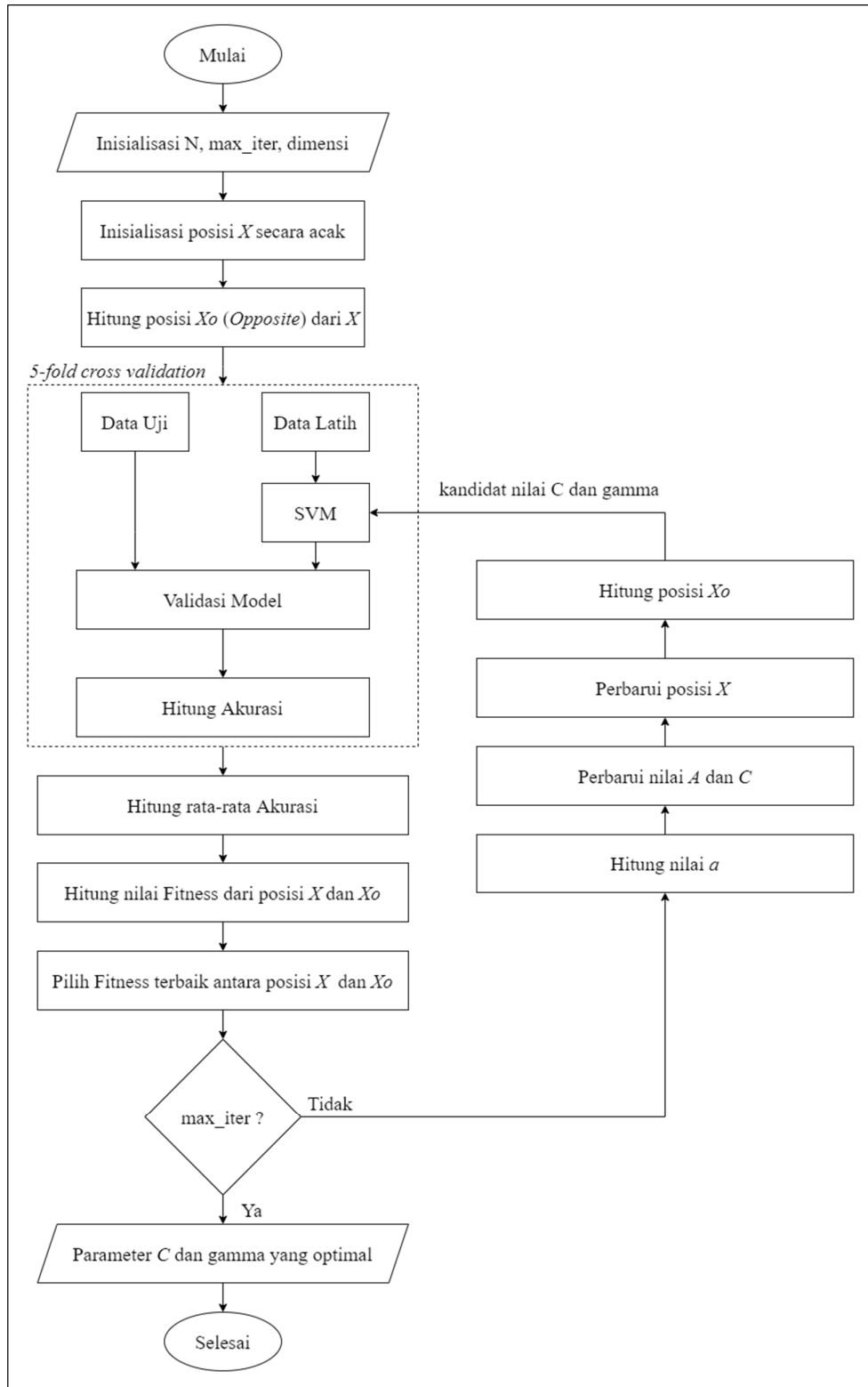
memilih *fitness* terbaik antara posisi X dan X_o untuk menghasilkan posisi (kandidat nilai C dan $gamma$) yang baru. Jika kondisi berhenti belum terpenuhi, maka hitung nilai a sesuai Persamaan (2-8), perbarui nilai A dan C sesuai persamaan (2-6)(2-7) , perbarui posisi X sesuai Persamaan (2-5) dan hitung nilai X_o berdasarkan persamaan (3-1) Selanjutnya, hitung kembali nilai *fitness*-nya dan pilih *fitness* terbaik antara posisi X dan X_o . Jika kondisi berhenti telah terpenuhi, maka posisi X atau X_o dengan nilai *fitness* terkecil menjadi parameter terbaik.

Hal penting yang sangat berpengaruh dalam algoritma SVM adalah pemilihan fungsi kernel dan parameternya. Oleh karena itu, penelitian ini akan menguji beberapa skenario pada Tabel 3.1 untuk mendapatkan kernel dan parameter yang optimal.

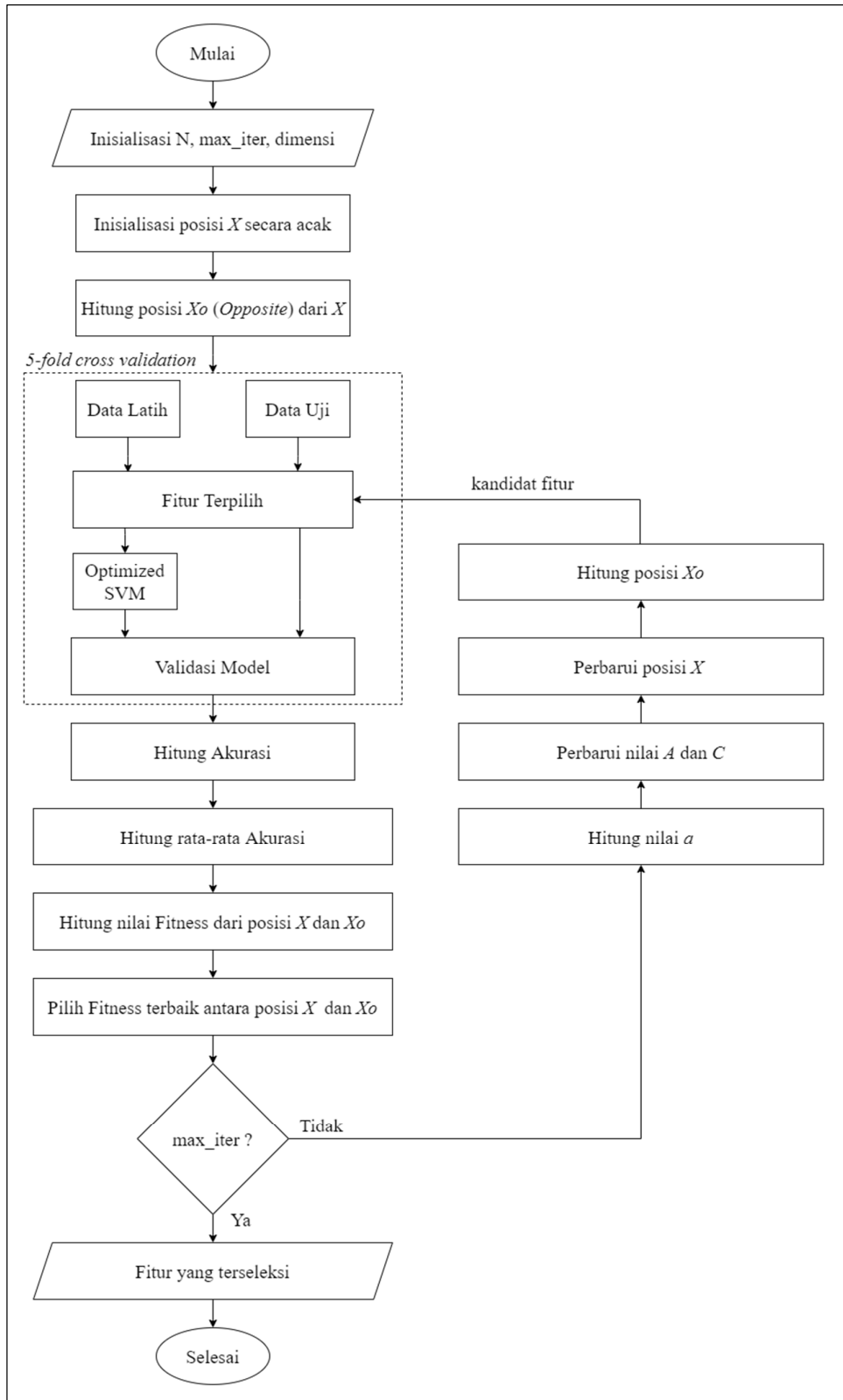
Tabel 3.1 Skenario Pengujian untuk Mendapatkan Parameter Optimal SVM

Kernel	C	gamma
Linear	0 – 10	N/A
RBF	0 – 10	0 – 10
Sigmoid	0 – 10	0 – 10
Polynomial	0 – 10	0 – 10

Setiap skenario tersebut akan dijalankan menggunakan 100 iterasi dengan jumlah populasi sebanyak 10. Sementara batas minimal (lb) nilai C dan $gamma$ adalah 0, dan batas maksimum (ub) nilai C dan $gamma$ adalah 10.



Gambar 3.2 Alur Proses Optimasi Parameter SVM Berbasis OBWOA



Gambar 3.3 Alur Proses Seleksi Fitur Menggunakan OBWOA

5. Seleksi Fitur

Proses pada seleksi fitur kurang lebih hampir sama dengan tahapan pada proses optimasi parameter SVM. Hanya saja fungsi *fitness* yang digunakan pada tahapan seleksi fitur ini mengikuti penelitian sebelumnya (Too & Mirjalili, 2021):

$$Fitness = \alpha ER + \beta \left(\frac{\#SF}{\#TF} \right)$$

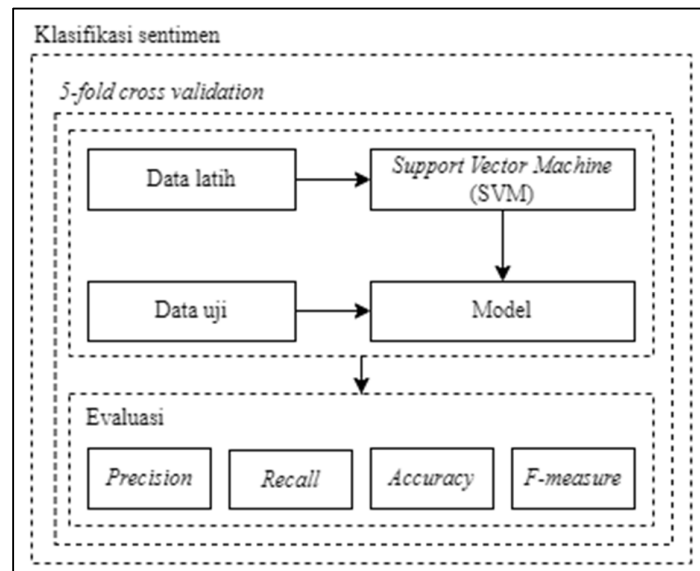
Dimana *ER* adalah nilai error klasifikasi, *#SF* adalah jumlah fitur yang terpilih, dan *#TF* adalah jumlah keseluruhan fitur. Nilai $\alpha \in [0,1]$ dan $\beta = (1 - \alpha)$ merupakan dua bobot yang dipakai untuk menunjukkan seberapa besar pengaruh nilai error klasifikasi dan jumlah fitur. Adapun proses kerja dari seleksi fitur berbasis OBWOA ini dapat dilihat pada Gambar 3.3.

Pada Gambar 3.3 proses seleksi fitur diawali dengan menginisialisasi parameter OBWOA seperti jumlah populasi (*N*), jumlah iterasi maksimum (*max_iter*), nilai batas atas (*ub*) dan nilai batas bawah (*lb*) serta dimensi atau jumlah fitur keseluruhan. Selanjutnya, nilai posisi awal *X* (kandidat fitur) dinisialisasi dan nilai *X_o* dihitung berdasarkan Persamaan (3-1) Setelah itu, dilakukan klasifikasi pada dataset yang divalidasi dengan menggunakan *5-fold cross validation*. Proporsi pembagian dataset adalah 80% untuk training dan 20% untuk testing. Algoritma SVM digunakan untuk mencari nilai rata-rata akurasi sebagai masukan pada fungsi *fitness*. Kemudian memilih *fitness* terbaik antara posisi *X* dan *X_o* untuk menghasilkan posisi (kandidat fitur) yang baru. Jika kondisi berhenti belum terpenuhi, maka hitung nilai *a* sesuai Persamaan (2-8), perbarui nilai *A* dan *C* sesuai Persamaan (2-6)(2-7), perbarui posisi *X* sesuai Persamaan (2-5) dan hitung nilai *X_o* berdasarkan Persamaan (3-1) Selanjutnya,

hitung kembali nilai *fitness*-nya dan pilih *fitness* terbaik antara posisi X dan X_0 . Jika kondisi berhenti telah terpenuhi, maka posisi X atau X_0 dengan nilai *fitness* terkecil menjadi fitur terbaik.

6. Klasifikasi Sentimen

Pada tahapan ini, algoritma yang akan digunakan untuk klasifikasi sentimen adalah *Support Vector Machine* (SVM). *Dataset* yang akan di klasifikasi ada dua, yaitu data hasil seleksi fitur dan tanpa seleksi fitur. Proses klasifikasi diawali dengan membagi *dataset* menjadi 80% data latih (*training*) dan 20% data uji (*testing*). Selanjutnya, algoritma SVM dijalankan untuk mengklasifikasikan sentimen. Model *machine learning* yang telah terbentuk lalu disimpan dalam format file *.pkl* menggunakan *library Pickle*. Alur dari tahapan klasifikasi sentimen dapat dilihat pada Gambar 3.4.



Gambar 3.4 Alur Klasifikasi Sentimen

7. Validasi dan Evaluasi

Pada tahap ini, algoritma *machine learning* divalidasi menggunakan *5-fold cross validation* dan dievaluasi menggunakan empat indikator pengukuran yaitu *accuracy*, *precision*, *recall* dan *f-measure*. Kemudian, waktu komputasi juga akan digunakan untuk menilai performa algoritma *machine learning*. Semakin kecil waktu komputasi maka performa algoritma tersebut semakin bagus.

8. Implementasi Model dan Visualisasi Hasil

Pada tahapan ini model terbaik yang telah dibangun, selanjutnya akan digunakan untuk mengklasifikasikan opini masyarakat yang diperoleh dari twitter. Kemudian hasil klasifikasi tersebut akan ditampilkan dalam bentuk diagram garis, diagram *pie*, dan *wordcloud*.

Diagram garis pada penelitian ini digunakan untuk melihat tren sentimen masyarakat berdasarkan waktu. Sedangkan diagram *pie* digunakan untuk menampilkan persentase sentimen Positif, Negatif, dan Netral berdasarkan data yang telah diklasifikasi.

Untuk visualisasi *wordcloud* digunakan untuk melihat kata-kata yang sering muncul dalam teks. Semakin sering kata tersebut muncul, maka semakin besar pula ukuran font yang ditampilkan dibanding kata-kata yang lain.

BAB 4 HASIL DAN PEMBAHASAN

Sebelum menjabarkan hasil sentimen tweet masyarakat, terlebih dahulu ditampilkan hasil pengumpulan data, eksplorasi data, preprocessing, ekstraksi fitur, optimasi parameter, seleksi fitur, validasi dan evaluasi pada model.

4.1 Pengumpulan dan Eksplorasi Data

Pada tahapan ini menghasilkan dua data yakni data yang bersumber dari Twitter dan Kaggle Dataset. Untuk dataset yang bersumber dari twitter digunakan sebagai data utama pada penelitian, sedangkan data pada kaggle digunakan untuk membangun model.

Jumlah keseluruhan dataset yang berhasil diperoleh dari twitter adalah sebanyak 1.389.481 baris dan 17 kolom, dengan rentan waktu selama 1 tahun (1 Januari 2021 s.d 31 Desember 2021). Pada Gambar 4.1 merupakan contoh hasil pengambilan data twitter dengan kata kunci yang digunakan adalah “PPKM”. Adapun atribut-atribut *tweet* yang diambil antara lain: *Datetime*, *Tweet Id*, *Text*, *Username*, *UserID*, *Location*, *UserCreated*, *Verified*, *Friends*, *Followers*, *TweetCount*, *Reply*, *Retweet*, *Like*, *Mention*, *Device*, dan *Media*.

Datetime	Tweet Id	Text	Username	UserID	Location	UserCreated	Verified	Friends	Followers	TweetCount	Reply	Retweet
2021-01-03 15:11:32	1.345750e+18	@sygkmajg Budakna wa endin bisi hayang apal im...	korban_ppkm	4.228994e+09	Garutjawa barat indonesia	2015-11-14 00:47:05	False	111	43	404	1	0
2021-01-03 15:14:08	1.345750e+18	@PRFMnews Sagala mahal, dahar jeng lalab we ti...	korban_ppkm	4.228994e+09	Garutjawa barat indonesia	2015-11-14 00:47:05	False	111	43	404	0	0
2021-01-03 16:35:59	1.345771e+18	@Bagoi62 @sygkmajg Hareupna aya solokan pokona mh	korban_ppkm	4.228994e+09	Garutjawa barat indonesia	2015-11-14 00:47:05	False	111	43	404	1	0

Gambar 4.1 Hasil Pengambilan Data dari Twitter

Berikut merupakan detail karakteristik dari data yang telah diambil dari twitter :

```

RangeIndex: 1389481 entries, 0 to 1389480
Data columns (total 17 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Datetime              1389481 non-null  datetime64[ns]
1   Tweet Id              1389481 non-null  float64
2   Text                  1389481 non-null  object
3   Username              1389481 non-null  object
4   UserID                1389481 non-null  float64
5   Location              901772 non-null   object
6   UserCreated           1389481 non-null  datetime64[ns]
7   Verified              1389481 non-null  bool
8   Friends               1389481 non-null  int64
9   Followers             1389481 non-null  int64
10  TweetCount            1389481 non-null  int64
11  Reply                 1389481 non-null  int64
12  Retweet               1389481 non-null  int64
13  Like                  1389481 non-null  int64
14  Mention               463394 non-null   object
15  Device                1389481 non-null  object
16  Media                 312617 non-null   object
dtypes: bool(1), datetime64[ns](2), float64(2), int64(6), object(6)
memory usage: 170.9+ MB

```

Gambar 4.2 Karakteristik Data Hasil *Scrapping*

Selanjutnya dilakukan pengambilan data dari Kaggle Dataset untuk dijadikan sebagai data latih dalam penelitian ini. Berikut adalah hasil pengambilan data dari Kaggle Dataset:

	class	comment
0	positif	Kami siap melaksanakan Instruksi pak
1	positif	Siap melaksanakan intruksi pak
2	positif	Siap dukung dan sukseskan
3	positif	Langkah 3M ini sudah sukses di Bali memutus pe...
4	positif	Siap amankan seluruh kebijakan kementerian Des...

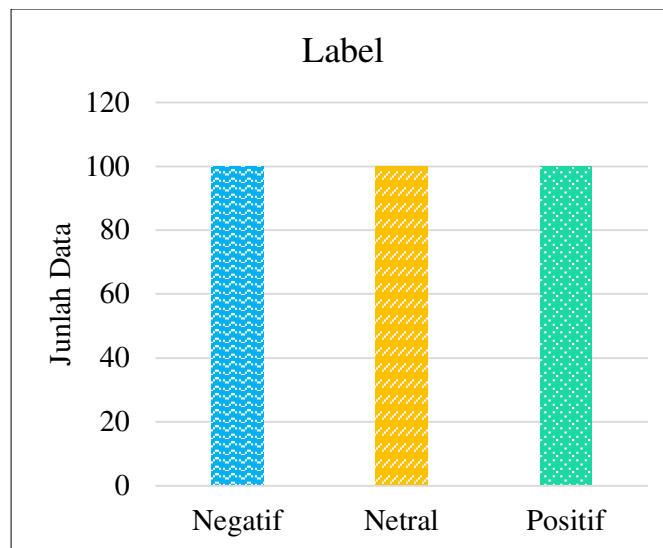
Gambar 4.3 Hasil Pengambilan Data dari Kaggle

Dari Gambar 4.3 dapat diketahui bahwa terdapat dua kolom, yang masing-masingnya menyimpan nilai *class* dan *comment*. Untuk *class* itu sendiri bermakna

sebagai label sentimen, sedangkan *comment* berisi opini masyarakat. Agar lebih jelasnya berikut ditampilkan karakteristik data latih yang diperoleh.

```
RangeIndex: 300 entries, 0 to 299
Data columns (total 2 columns):
#   Column   Non-Null Count  Dtype
---  -
0   class    300 non-null    object
1   comment  300 non-null    object
dtypes: object(2)
memory usage: 4.8+ KB
```

Gambar 4.4 Karakteristik Data Latih



Gambar 4.5 Distribusi Jumlah setiap Label

Berdasarkan Gambar 4.4 dapat dilihat bahwa jumlah data secara keseluruhan adalah sebanyak 300 baris dan 2 kolom, dimana khusus untuk kolom *class* menyimpan 3 label, yakni positif, negatif, dan netral yang jumlah tiap labelnya dapat dilihat pada Gambar 4.5.

Dari Gambar 4.5 dapat dilihat bahwa jumlah label pada data adalah masing-masing sebanyak 100 baris. Hal tersebut mengindikasikan bahwa label pada data memiliki distribusi yang seimbang (*balance*).

Setelah data yang dibutuhkan berhasil diperoleh, maka langkah selanjutnya adalah melakukan preprocessing pada masing-masing dataset.

4.2 Preprocessing

Data yang telah dikumpulkan merupakan data yang tidak terstruktur dengan baik. Berdasarkan hal tersebut, teknik preprocessing dilakukan agar data menjadi lebih baik dan dapat diolah dengan mudah oleh algoritma machine learning. Teknik ini diterapkan pada masing-masing dataset dengan tahapan yang meliputi filtering, casefolding, slangword, stopword, stemming dan tokenization. Hasil dari masing-masing tahapan *preprocessing* dapat dilihat pada Tabel 4.1.

Berdasarkan Tabel 4.1 teknik preprocessing diawali dengan tahapan filtering yaitu menghapus URL, menghapus tanda @USER, #Hashtag, Retweet, tanda baca, angka dan karakter lainnya serta mengubah setiap kata menjadi huruf kecil (casefolding). Selanjutnya, tahapan slangword yaitu setiap kata yang tidak baku diubah menjadi kata baku agar kata tersebut seragam dan memiliki makna yang sama, seperti kata “nggak” menjadi “tidak”. Berikutnya, menghapus kata-kata yang tidak penting (stopword) dan tidak memiliki hubungan pada analisis sentimen, seperti menghapus kata “yang”, “lagi”, “sampai”, “kalau”, “ini”, “pun”, “di”, “dengan”. Untuk mengubah kata menjadi bentuk kata dasarnya maka dilakukan stemming yaitu menghilangkan awalan dan akhiran kata, seperti kata “berkerumun” menjadi “kerumun”, “pidanakan” menjadi “pidana”, dan “pelanggaran” menjadi

“langgar”. Tahapan terakhir dari preprocessing adalah tokenization, yaitu memisahkan kalimat menjadi beberapa kata.

Tabel 4.1 Hasil Tahapan Preprocessing

Tahap	Teks
Teks awal	Hasil PPKM yang nggak jelas... Tutup jam 8, buka lagi jam 10 sampai pagi... berkerumun! Aneh kalau ini pun tidak dipidanakan dengan pelanggaran prokes!! https://t.co/jXCWlxtyKZ
Filtering	Hasil PPKM yang nggak jelas Tutup jam buka lagi jam sampai pagi berkerumun Aneh kalau ini pun tidak di pidanakan dengan pelanggaran prokes
Casefolding	hasil ppkm yang nggak jelas tutup jam buka lagi jam sampai pagi berkerumun aneh kalau ini pun tidak di pidanakan dengan pelanggaran prokes
Slangword	hasil ppkm yang tidak jelas tutup jam buka lagi jam sampai pagi berkerumun aneh kalau ini pun tidak di pidanakan dengan pelanggaran prokes
Stopword	hasil ppkm tidak jelas tutup jam buka jam pagi berkerumun aneh tidak pidanakan pelanggaran prokes
Stemming	hasil ppkm tidak jelas tutup jam buka jam pagi kerumun aneh tidak pidana langgar prokes
Tokenization	['hasil', 'ppkm', 'tidak', 'jelas', 'tutup', 'jam', 'buka', 'jam', 'pagi', 'kerumun', 'aneh', 'tidak', 'pidana', 'langgar', 'prokes']

4.3 Ekstraksi Fitur

Data yang telah melalui tahapan preprocessing, selanjutnya diubah dari teks menjadi vektor agar dapat diproses oleh algoritma machine learning. Pada penelitian ini, proses ekstraksi fitur atau pembobotan kata menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Pada metode TF-IDF, bobot akan semakin besar apabila frekuensi kemunculan kata pada suatu dokumen semakin tinggi. Akan tetapi, bobot akan berkurang apabila kata tersebut semakin

sering muncul pada dokumen lainnya. Dari hasil ekstraksi fitur ini, didapatkan jumlah fitur sebanyak 964. Berikut merupakan hasil dari tahapan ekstraksi fitur:

	aamin	abai	abdi	absurd	ade	adik	adil	adt	agama	ah	...	wita	wujud	yak	yasir	yang	yogyakarta	your	yra	yustisi	zona		
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...
237	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
238	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
239	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
240	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.345232	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
241	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

242 rows x 964 columns

Gambar 4.6 Hasil Ekstraksi Fitur

4.4 Hasil dan Analisis Optimasi Parameter SVM

Dalam penelitian ini, Support Vector Machine (SVM) adalah algoritma machine learning yang digunakan untuk mengklasifikasikan sentimen. Namun, ada dua tantangan dalam SVM, seperti pemilihan parameter yang optimal dan pemilihan fungsi kernel. Oleh karena itu, OBWOA digunakan untuk mengoptimalkan parameter SVM sesuai dengan topik permasalahan yang dihadapi.

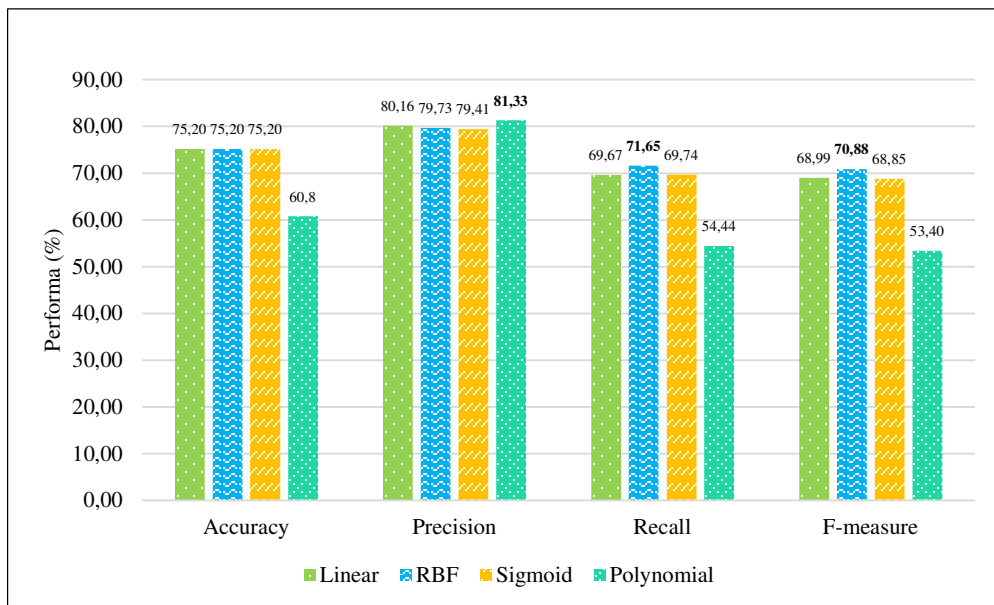
Tabel 4.2 Hasil Pengujian Optimasi Parameter SVM

Kernel	C	gamma	Parameter Optimal
Linear	0 – 10	N/A	$C = 0.52519819$
RBF	0 – 10	0 – 10	$C = 4.99522643$, $gamma = 1.4236435$
Sigmoid	0 – 10	0 – 10	$C = 9.76346309$, $gamma = 0.05604261$
Polynomial	0 – 10	0 – 10	$C = 1.90727318$, $gamma = 4.0498844$

Pada eksperimen ini, *5-fold cross validation* digunakan sebagai validator dengan metrik performa yang diukur antara lain *accuracy*, *precision*, *recall*, dan *f-measure*. Jumlah pembagian data untuk training adalah sebesar 80% dan testing

20%. Proporsi ini dipilih dengan menyesuaikan jumlah dataset yang diperoleh dari Kaggle Dataset. Adapun penentuan parameter dan pemilihan fungsi kernel pada SVM dilakukan berdasarkan skenario dengan rentang nilai C dan gamma masing-masing 0 hingga 10. Skenario beserta nilai hasil optimasi ditampilkan pada Tabel 4.2.

Setiap skenario tersebut masing-masing dijalankan pada algoritma OBWOA dengan jumlah populasi sebanyak 10 dan maksimum 100 iterasi. Hasil performa ditunjukkan pada Gambar 4.7.



Gambar 4.7 Hasil Performa Pengujian Optimasi SVM

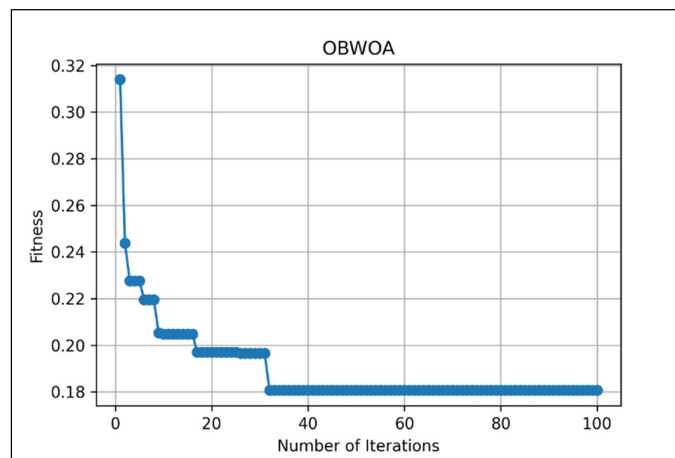
Pada Gambar 4.7 dapat dilihat bahwa metrik *accuracy* dari masing-masing kernel adalah sama yaitu 75,20%, kecuali Polynomial yang hanya 60,8%. Untuk metrik *precision*, kernel Polynomial memiliki nilai tertinggi yaitu 81,33% kemudian Linear 80,16%, RBF 79,73%, lalu Sigmoid 79,41%. Pada metrik *recall* dan *f-measure*, kernel RBF berhasil mengungguli kernel lainnya dengan nilai masing-masing 71,65% dan 70,88%.

Maka berdasarkan hasil optimasi tersebut, kernel RBF dengan nilai $C = 4.99522643$ dan $gamma = 1.4236435$ dipilih untuk digunakan pada eksperimen seleksi fitur karena memiliki performa yang baik hampir di setiap metrik pengukuran.

4.5 Hasil dan Analisis Seleksi Fitur

Berikutnya, penelitian ini membandingkan hasil klasifikasi sentimen dengan menggunakan seleksi fitur dan tanpa seleksi fitur. Masing-masing proses divalidasi menggunakan *5-fold cross validation* dan metrik yang diukur adalah *accuracy*, *precision*, *recall*, dan *f-measure*. Namun sebelum dilakukan klasifikasi, terlebih dahulu ditampilkan hasil seleksi fitur menggunakan algoritma OBWOA.

Berikut ini merupakan kurva konvergensi nilai fitness dari seleksi fitur menggunakan kombinasi OBWOA dengan SVM yang telah dioptimasi.

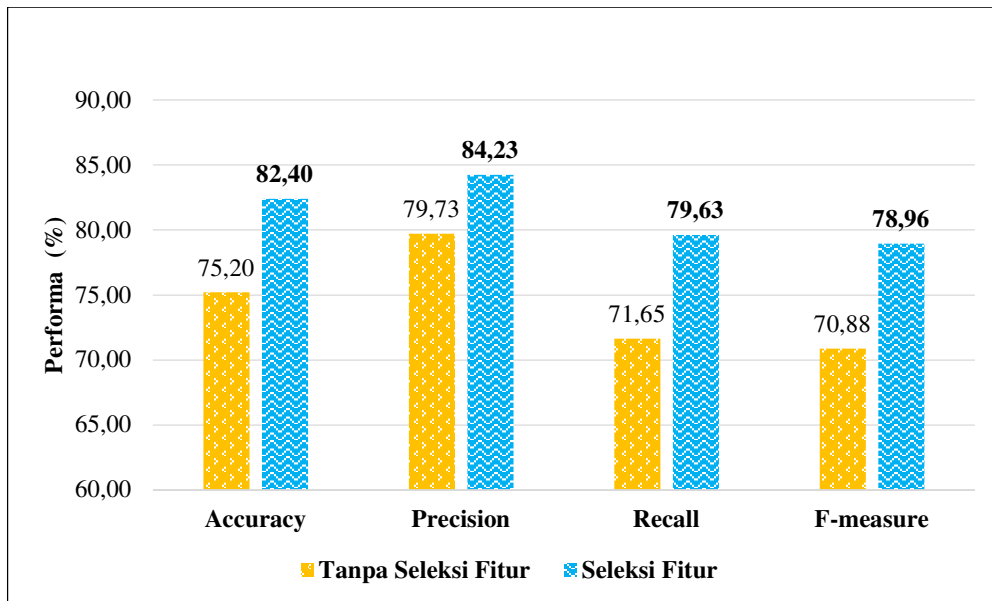


Gambar 4.8 Kurva Konvergensi Seleksi Fitur

Dari Gambar 4.8 perlu diketahui bahwa proses seleksi fitur ini menggunakan 100 iterasi dan jumlah populasi sebanyak 10. Pada kurva tersebut, OBWOA mencapai fitness terendahnya (konvergen) pada iterasi 30 hingga 100.

Berdasarkan eksperimen ini, fitur yang berhasil terseleksi menggunakan OBWOA adalah sebanyak 628 fitur dari 964 total keseluruhan fitur.

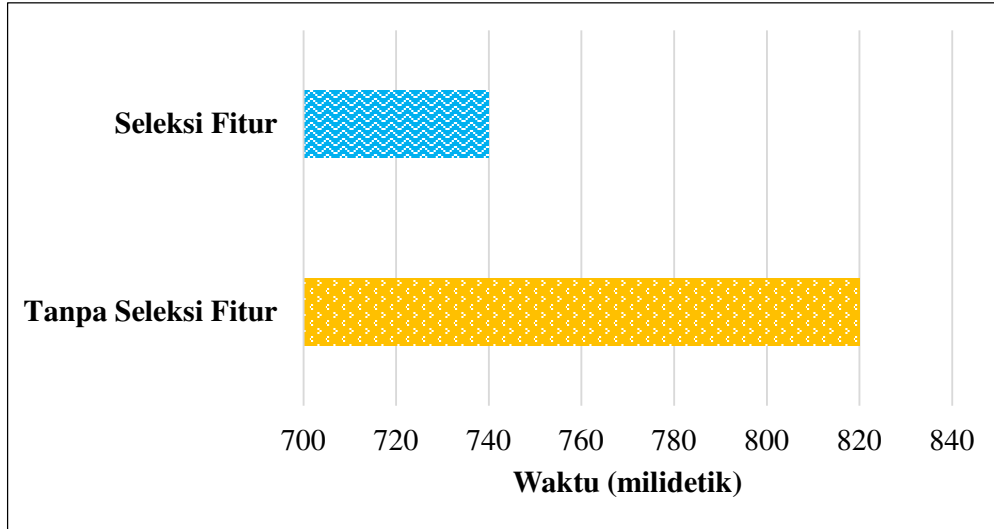
Selanjutnya, klasifikasi menggunakan seleksi fitur dibandingkan performanya dengan klasifikasi tanpa seleksi fitur sebagaimana ditampilkan pada Gambar 4.9. Dari Gambar 4.9 dapat diketahui bahwa dengan menerapkan seleksi fitur, performa klasifikasi dapat meningkat secara signifikan. Hal ini ditandai dengan nilai *accuracy*, *precision*, *recall*, dan *f-measure* yang masing-masing meningkat sebesar 7,20%, 4,50%, 7,98%, dan 8,08% dari klasifikasi tanpa menggunakan seleksi fitur.



Gambar 4.9 Perbandingan Hasil Performa Seleksi Fitur

Kemudian, penelitian ini juga membandingkan waktu komputasi ketika menggunakan seleksi fitur dan tanpa seleksi fitur. Hasil pengujian menunjukkan bahwa klasifikasi dengan menggunakan seleksi fitur membutuhkan 80 milidetik lebih cepat dari klasifikasi tanpa menggunakan seleksi fitur. Hal ini membuktikan bahwa metode seleksi fitur bukan hanya dapat meningkatkan performa klasifikasi

sentimen tapi juga mampu mengurangi waktu komputasi. Hasilnya dapat dilihat pada Gambar 4.10.

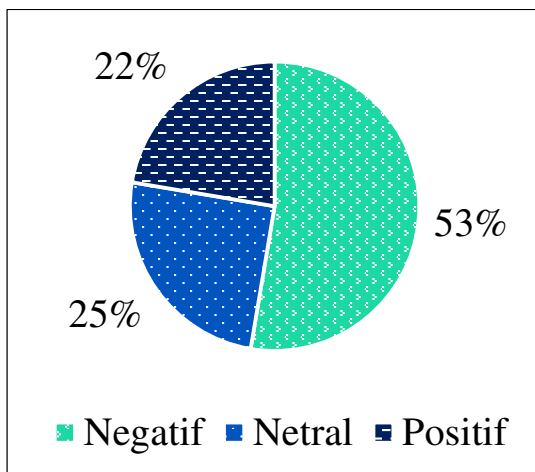


Gambar 4.10 Perbandingan Waktu Komputasi

4.6 Hasil dan Analisis Sentimen Masyarakat terhadap Kebijakan PPKM

Penelitian ini berhasil menghimpun (*scrapping*) sebanyak 1.389.481 tweet di Twitter terkait respon masyarakat terhadap kebijakan PPKM dengan menggunakan library *snsrape* dari tanggal 01 Januari 2021 hingga 31 Desember 2021. Data tersebut diproses terlebih dahulu untuk kemudian dilakukan klasifikasi sentimen menggunakan model machine learning yang didapatkan dari hasil eksperimen sebelumnya.

Hasil klasifikasi sentimen untuk data tersebut ditampilkan pada Gambar 4.11. Berdasarkan Gambar 4.11 terdapat 731.761 tweet atau sekitar 53% bersentimen Negatif, 347.380 tweet atau sekitar 25% bersentimen Netral, dan 310.340 tweet atau sekitar 22% bersentimen Positif. Contoh tweet dengan masing-masing sentimen dapat dilihat pada Gambar 4.12.



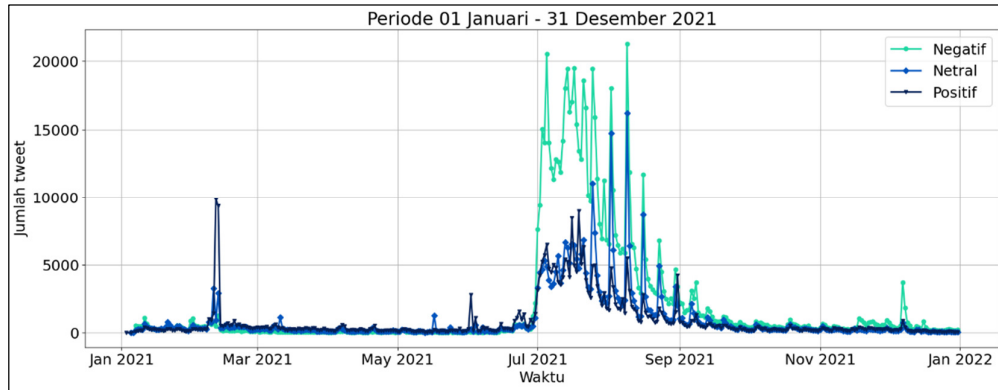
Gambar 4.11 Hasil Sentimen PPKM

Berdasarkan contoh pada Gambar 4.12, dapat kita lihat bahwa tweet dengan sentimen Negatif berisi kritikan, sindiran hingga keluhan masyarakat selama program PPKM. Lalu, tweet dengan sentimen Netral berasal dari akun-akun berita dan hanya menyinggung soal pelaksanaan program PPKM. Sedangkan tweet dengan sentimen Positif berisi ajakan dan dukungan terhadap pemerintah untuk menyukseskan kebijakan PPKM.

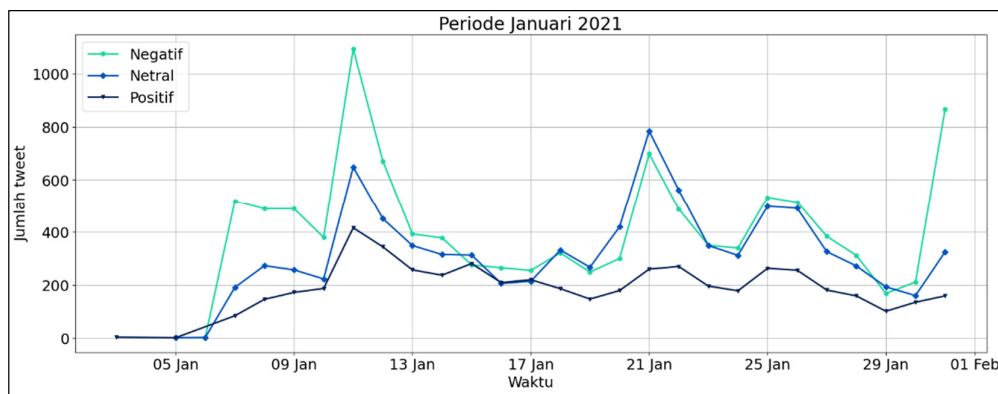
Negatif	Netral	Positif
<p>YOS @onepebudianto Ternyata banyak yang harus dihafal ya kalo hidup di Indonesia 🤔😓 Setelah PSBB, PSBB Transisi, PSBB Ketat, PSBK, muncul PPKM. Bakal ada apa lagi yak 🤔😓😓 Jangan cuma jago bikin aturan dan istilah. Implementasi di lapangan juga harus garang boskuuu 🤨</p> <p>3:44 PM - Jan 10, 2021 - Twitter for Android</p>	<p>Merdeka.com @merdekadotcom Jelang PPKM Level 4 Berakhir, Lalu Lintas di Sejumlah Pos Penyekeatan DKI Ramai Lancar: Arus lalu lintas terpantau ramai lancar di sejumlah pos penyekeatan di beberapa titik pada masa Pemberlakuan pembatasan kegiatan masyarakat (PPKM) Level 4 di Jakarta... dvr.it/S4pjgc</p> <p>11:17 AM - Aug 1, 2021 - dvr.it</p>	<p>widodo7908 @widodo7908 Replying to @poldajateng... PPKM untuk kesehatan bersama Ayo dukung Pemerintah terapkan PPKM #AyoDukungPPKM</p> <p>11:38 AM - Feb 10, 2021 - Twitter for Android</p>
<p>arlyebrian @arlyebrian Replying to @detikcom Emang tuh @jokowi slalu nyari jalan tengah pebb ppcm level levelan..akibatnya malah lbh buruk rugi besar..anggaran abisnya ratusan triliun tp tetap angka covid bnyak rakyat tetap sengsara km dampak ekonomi..dpt bantuan cmn dikit..pejabatnya dn aparat aja yg kenyang!</p> <p>8:55 AM - Aug 1, 2021 - Twitter for Android</p>	<p>LIPUTANS @liputan6dotcom Pengamat Prediksi PPKM Level akan Diperpanjang tapi Ada Kelonggaran</p> <p>liputan6.com Pengamat Predikal PPKM Level akan Diperpanjang tapi Ada... PPKM level diberlakukan untuk membatasi mobilitas masyarakat demi mencegah penyebaran Covid-19.</p> <p>12:31 PM - Aug 1, 2021 - dvr.it</p>	<p>Sumarjoamenda@gmail.com @sumarjoamenda Replying to @Polresta_Solo Monggo sedulur warga masyarakat Solo dukung dan patuhi PPKM dengan tetap memakai masker hindari kerumunan cuci tangan dan jago jarak. Biar kita semua cepet sehat. @poldajateng @PolrestaS #polisidatangmasyarakatsewang</p> <p>12:22 PM - Feb 10, 2021 - Twitter for Android</p>
<p>Ibu Talin @bakactarhari Parameter dan indikator apa yg menjadi tolak ukur keberhasilan pemerintah dim penanganan pandemi jangan dilihat perpanjangan tp lihat dr kebijakan pertama Ppkm darurat sampai skrg Sepertinya hasil yg didapatkan tidak sebanding dengan pengorbanan rakyat yg sudah menjerit</p> <p>Translate Tweet</p>	<p>TMC Polda Metro Jaya @TMC PoldaMetro 08.24 Polri Sat Lantas Jakbar melaksanakan Penyekeatan arus lalu lintas dalam rangka Pemberlakuan Pembatasan Kegiatan Masyarakat Level 4 untuk pencegahan penyebaran Covid-19 di Pos PPKM Kalideres Jakarta Barat.</p> <p>Translate Tweet</p>	<p>Cantika Putri @caanputri Beberapa daerah cukup berhasil menurunkan angka penyebaran Covid-19 & bed rate di RS dg menerapkan PPKM. Efektivitas kebijakan ini bisa ditiru daerah lain sehingga kasus positif di Indonesia makin berkurang. m.id/baca-berita/na... Vaksin Untuk Semua #PPKMEfektifTurunkanCovid</p> <p>Translate Tweet</p>

Gambar 4.12 Contoh Tweet Sentimen Negatif, Netral, dan Positif

Berikutnya, ditampilkan visualisasi tren sentimen mengenai PPKM pada Gambar 4.13. Eksplorasi tren sentimen dilakukan untuk melihat seberapa banyak tweet mengenai PPKM dibicarakan dalam waktu tertentu.



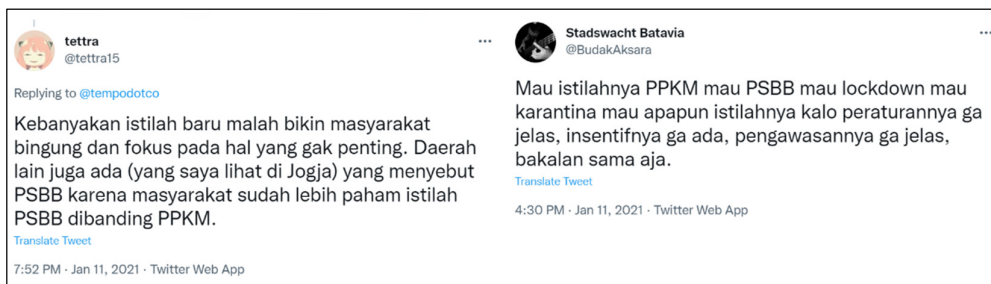
Gambar 4.13 Tren Sentimen Berdasarkan Waktu



Gambar 4.14 Tren Sentimen Bulan Januari 2021

Pada Gambar 4.14, merupakan tren sentimen pada periode Januari 2021. Pada grafik tersebut, jumlah sentimen terlihat fluktuatif. Meski ada beberapa hari dimana tweet dengan sentimen Negatif mendominasi, ada pula hari ketika tweet dengan sentimen Netral lebih dominan. Misalnya pada tanggal 11 Januari jumlah sentimen Negatif mencapai lebih dari 1000 tweet. Hal ini terjadi karena pada tanggal tersebut merupakan hari dimana kebijakan PPKM pertama kali diberlakukan dan kebanyakan masyarakat bingung terhadap kebijakan pemerintah tersebut yang

dianggap hanya mengganti istilah penamaan. Beberapa contoh tweet dengan sentimen Negatif ditampilkan pada Gambar 4.15. Kemudian tepat 10 hari setelah kebijakan PPKM diberlakukan yakni pada tanggal 21 Januari terjadi lonjakan tweet dengan sentimen Netral dan Negatif. Lonjakan sentimen Netral ini terjadi karena banyak tweet yang didominasi berita terkait isu perpanjangan kebijakan PPKM, sementara sentimen Negatif disebabkan karena kasus COVID di Indonesia yang terus bertambah meskipun telah diterapkan PPKM. Sehingga masyarakat beranggapan bahwa pemerintah kurang tegas dalam menerapkan kebijakan ini. Beberapa contoh tweet dengan sentimen Netral ditampilkan pada Gambar 4.16.



Gambar 4.15 Contoh Tweet Sentimen Negatif di Bulan Januari

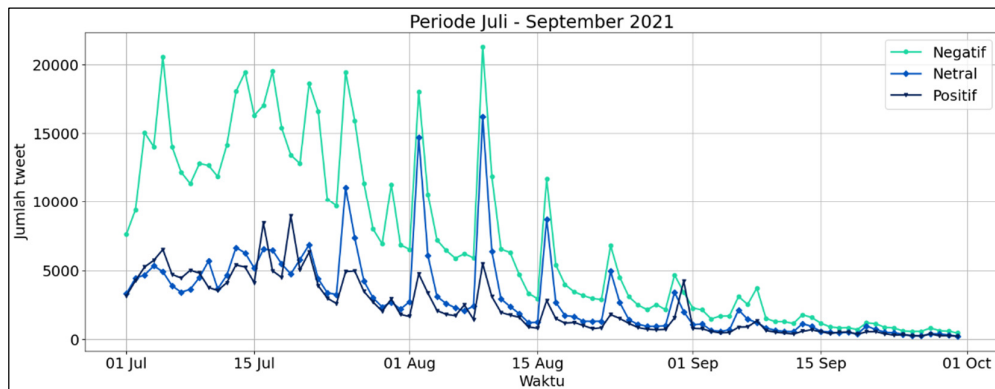


Gambar 4.16 Contoh Tweet Sentimen Netral di Bulan Januari

Berikutnya pada pertengahan bulan Februari terlihat ada lonjakan tren sentimen Positif seperti yang terlihat pada Gambar 4.17.

“landai” . Hal tersebutlah yang menjadi penyebab tren sentimen Positif lebih mendominasi dibandingkan sentimen Negatif dan Netral.

Selanjutnya, pada awal Juli hingga akhir September terjadi lonjakan tren sentimen Negatif yang grafiknya dapat dilihat pada gambar berikut.



Gambar 4.19 Tren Sentimen Bulan Juli – September 2021

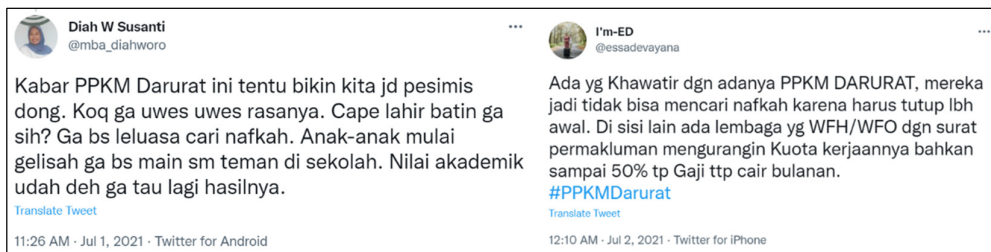


Gambar 4.20 Wordcloud Sentimen Negatif Bulan Juli - September

Berdasarkan Gambar 4.19 dapat dilihat bahwa tren sentimen Negatif mencapai lebih dari 20.000 tweet pada awal Juli dan puncaknya terjadi pada bulan Agustus. Hal tersebut terjadi karena pada bulan ini, kasus positif dan jumlah kematian akibat Covid-19 di Indonesia meningkat drastis disebabkan oleh penularan varian baru

virus Corona yaitu Varian Delta. Untuk melihat topik yang paling banyak di bicarakan masyarakat di twitter, maka ditampilkan wordcloud untuk sentimen Negatif pada bulan Juli.

Pada Gambar 4.20 dapat diketahui bahwa, topik yang paling sering diperbincangkan masyarakat adalah PPKM Darurat, dimana dampak dari PPKM Darurat ini meliputi pembatasan aktivitas masyarakat yang lebih ketat daripada kebijakan sebelumnya. Misalnya pada contoh tweet Gambar 4.21, masyarakat beranggapan bahwa dengan diberlakukannya PPKM Darurat membuat mereka merasa kesulitan dalam mencari nafkah dan mendapatkan penghasilan karena semua diharuskan untuk bekerja dari rumah (WFH).



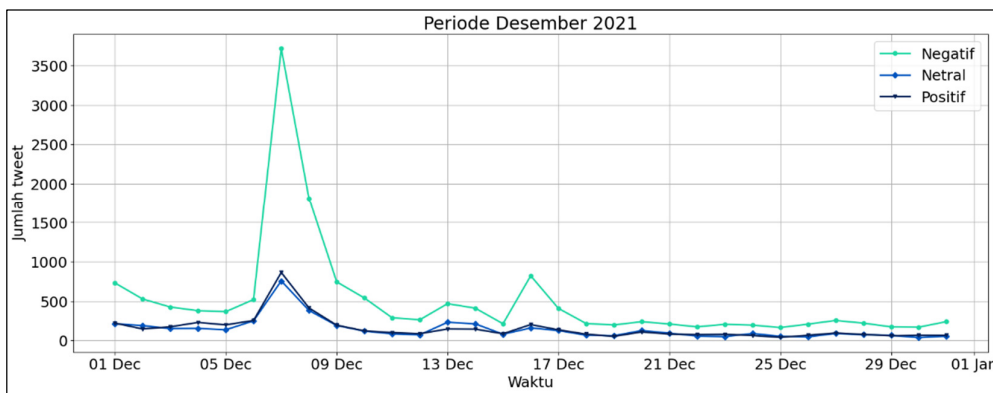
Gambar 4.21 Contoh Tweet Sentimen Negatif di Bulan Juli

Kemudian sentimen Negatif berangsur menurun pada awal hingga akhir September. Melihat penurunan sentimen Negatif ini, bisa dikatakan bahwa masyarakat mulai mendukung kebijakan pemerintah terkait PPKM dalam upaya menangani pandemi Covid-19. Hal ini juga membuktikan bahwa penerapan PPKM sukses menekan kasus positif Covid-19, sehingga pemerintah mulai melonggarkan beberapa aturan PPKM. Dari Gambar 4.22. menunjukkan tweet masyarakat dengan sentimen Positif yang menyambut baik adanya pelonggaran PPKM.



Gambar 4.22 Contoh Tweet Sentimen Positif di Bulan September

Dan terakhir terjadi lonjakan sentimen Negatif pada awal hingga puncaknya pada 07 Desember seperti yang terlihat pada Gambar 4.23. Pada awal Desember masyarakat mengomentari rencana pemerintah untuk memberlakukan PPKM saat Natal dan Tahun Baru. Namun, berselang beberapa saat, pemerintah kemudian mengumumkan untuk membatalkan kebijakan tersebut.



Gambar 4.23 Tren Sentimen Bulan Desember 2021

Hal inilah yang membuat sebagian besar masyarakat beranggapan bahwa pemerintah tidak konsisten dalam memutuskan kebijakan, seperti yang ditunjukkan pada Gambar 4.24.

Muthi Aghnia
@muthiaghnia

Gimana sih ga konsisten, udah bener mending PPKM aja. Mencegah lebih baik daripada mengobati kan?

[Translate Tweet](#)

Nuice Media @nuicemedia · Dec 7, 2021

LBP has canceled PPKM level 3 between 24 December and 2 January nationally in favor of determining the level according to weekly testing and vaccination efforts as it has been since July.

Vax 1 rate in Java-Bali has reached 76% and vax 2 nearing 56%. Nationally it's 68% and 47%

9:33 AM · Dec 7, 2021 · Twitter Web App

Bona ni Pinasa
@Defree

Sebenarnya sebelum melontarkan aturan dipikir dan dianalisa dulu gak sih? katanya akan ppcm, skrg gak jadi level anu.. kan dibayar bukan mengganggu aktifitas orang di akhir tahun. Gak semua orang bisa seenaknya mengubah jadwal setiap detik pak

[@jokoanwar @jokowi @Kemendagri_RI](#)

[Translate Tweet](#)

10:03 AM · Dec 7, 2021 · Twitter Web App

Adipati Djojonegoro™
@Dhox_TEEMGE

MISS KORDINASI antar lembaga kementrian. Sebelumnya @kemenkopmk @kemendagri terapkan PPKM level 3, selang 2 Minggu kemudian dibatalkan.

Ada apa sebenarnya ?? Lain kali klo buat kebijakan yg kompak.L, biar rakyat ini gak bingung dan pemerintah dianggap mencla mencle CC : @jokowi

[Translate Tweet](#)

moch haris
@mochhar74119177

Replying to @liputan6dotcom

Katanya waspada omicron, sekarang malah PPKM dibatalkan. Pemerintahan sekarang memang ga bisa dipercaya. Mungkin PPKM hanya berlaku saat peringatan atau perayaan hari besar umat Islam.

[Translate Tweet](#)

10:02 AM · Dec 7, 2021 · Twitter for Android

Gambar 4.24 Contoh Tweet Sentimen Negatif di Bulan Desember

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini telah dilakukan pengujian dengan menerapkan algoritma OBWOA pada klasifikasi SVM untuk optimasi parameter dan seleksi fitur. Selain itu, pada penelitian ini juga telah dilakukan klasifikasi terhadap data Twitter untuk mengetahui sentimen masyarakat terhadap kebijakan PPKM. Berdasarkan hal tersebut, maka dapat disimpulkan bahwa:

1. Hasil optimasi menggunakan algoritma OBWOA diperoleh parameter SVM terbaik dengan nilai $C = 4.99522643$ dan $\gamma = 1.4236435$. Hal ini ditunjukkan dengan performa klasifikasi yang baik dengan nilai rata-rata accuracy 75,20%, precision 79,73%, recall 71,65%, dan f-measure 70,88%.
2. Hasil pengujian seleksi fitur menggunakan algoritma OBWOA diperoleh performa klasifikasi yang baik dengan rata-rata accuracy, precision, recall, dan f-measure masing-masing sebesar 82,40%, 84,23%, 79,63%, dan 78,96% serta jumlah fitur yang berhasil terseleksi adalah sebanyak 652 fitur. Selain itu, metode seleksi fitur yang diusulkan juga mampu mengurangi waktu komputasi.
3. Hasil klasifikasi sentimen terhadap 1.389.481 data tweet masyarakat pada periode Januari hingga Desember 2021 diperoleh sebanyak 53% tweet dengan sentimen Negatif, 25% tweet sentimen Netral, dan 22% tweet sentimen Positif. Dari hasil ini dapat diketahui bahwa sentimen masyarakat terhadap kebijakan PPKM cenderung Negatif.

5.2 Saran

Adapun saran pada penelitian ini antara lain:

1. Pada penelitian ini, jumlah dataset latih yang digunakan masih sedikit.
Untuk penelitian selanjutnya, peneliti dapat menambah dataset agar performa klasifikasi dapat meningkat.
2. Pada penelitian selanjutnya, dapat mencoba menggunakan algoritma optimasi dan seleksi fitur yang lain.

DAFTAR PUSTAKA

- Abd Elaziz, M., & Oliva, D. (2018). Parameter estimation of solar cells diode models by an improved opposition-based whale optimization algorithm. *Energy Conversion and Management*, *171*(September 2017), 1843–1859. <https://doi.org/10.1016/j.enconman.2018.05.062>
- Delimayanti, M. K., Sari, R., Laya, M., Faisal, M. R., & ... (2021). Pemanfaatan Metode Multiclass-SVM pada Model Klasifikasi Pesan Bencana Banjir di Twitter. *Edu Komputika* ..., June. <https://doi.org/10.15294/edukomputika.v8i1.47858>
- Devika, M. D., Sunitha, C., & Ganesh, A. (2016). Sentiment Analysis: A Comparative Study on Different Approaches. *Procedia Computer Science*, *87*, 44–49. <https://doi.org/10.1016/j.procs.2016.05.124>
- Gulati, K., Saravana Kumar, S., Sarath Kumar Boddu, R., Sarvakar, K., Kumar Sharma, D., & Nomani, M. Z. M. (2021). Comparative analysis of machine learning-based classification models using sentiment classification of tweets related to COVID-19 pandemic. *Materials Today: Proceedings*, May. <https://doi.org/10.1016/j.matpr.2021.04.364>
- Hassouneh, Y., Turabieh, H., Thaher, T., Tumar, I., Chantar, H., & Too, J. (2021). Boosted Whale Optimization Algorithm with Natural Selection Operators for Software Fault Prediction. *IEEE Access*, *9*(January), 14239–14258. <https://doi.org/10.1109/ACCESS.2021.3052149>
- Jayani, D. H. (2021). *PPKM Darurat Diberlakukan, Aktivitas Masyarakat Dibatasi*. Katadata.Co.Id. <https://katadata.co.id/ariayudhistira/infografik/60dd9d3271f8e/ppkm-darurat->

diberlakukan-aktivitas-masyarakat-dibatasi

- Jianqiang, Z., & Xiaolin, G. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, 5, 2870–2879. <https://doi.org/10.1109/ACCESS.2017.2672677>
- JustAnotherArchivist. (2018). *snsrape: A social networking service scraper in Python* (0.3.5). <https://github.com/JustAnotherArchivist/snsrape>
- Kumar, A. (2020). *Data Science:Support Vector Machine(SVM)*. Medium.Datadriveninvestor.Com. <https://medium.datadriveninvestor.com/data-science-support-vector-machine-svm-fa74fb12d349>
- Laurensz, B., & Sedyono, E. (2021). Analisis Sentimen Masyarakat terhadap Tindakan Vaksinasi dalam Upaya Mengatasi Pandemi Covid-19. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 10(2), 118–123.
- Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Muhidin, D., & Wibowo, A. (2020). Perbandingan Kinerja Algoritma Support Vector Machine dan K-Nearest Neighbor Terhadap Analisis Sentimen Kebijakan New Normal. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 5(2), 153. <https://doi.org/10.30998/string.v5i2.6715>
- Nanda, M. A., Seminar, K. B., Nandika, D., & Maddu, A. (2018). A comparison study of kernel functions in the support vector machine and its application for termite detection. *Information (Switzerland)*, 9(1). <https://doi.org/10.3390/info9010005>

- Prastyo, P., Ardiyanto, I., & Hidayat, R. (2020). *A Review of Feature Selection Techniques in Sentiment Analysis Using Filter, Wrapper, or Hybrid Methods*.
- Prastyo, P. H., Ardiyanto, I., & Hidayat, R. (2020). Indonesian Sentiment Analysis: An Experimental Study of Four Kernel Functions on SVM Algorithm with TF-IDF. *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, ICDABI 2020*.
<https://doi.org/10.1109/ICDABI51230.2020.9325685>
- Prastyo, P. H., Hidayat, R., & Ardiyanto, I. (2022). Enhancing sentiment classification performance using hybrid Query Expansion Ranking and Binary Particle Swarm Optimization with Adaptive Inertia Weights. *ICT Express*, 8(2), 189–197. <https://doi.org/10.1016/j.ict.2021.04.009>
- Prastyo, P. H., Sumi, A. S., Dian, A. W., & Permanasari, A. E. (2020). Tweets Responding to the Indonesian Government's Handling of COVID-19: Sentiment Analysis Using SVM with Normalized Poly Kernel. *Journal of Information Systems Engineering and Business Intelligence*, 6(2), 112. <https://doi.org/10.20473/jisebi.6.2.112-122>
- Ramadhanny, F. (2021). *Pro Kontra Netizen Soal PPKM Darurat di Twitter*. DetikInet. <https://inet.detik.com/cyberlife/d-5627526/pro-kontra-netizen-soal-ppkm-darurat-di-twitter>
- Ruz, G. A., Henríquez, P. A., & Mascareño, A. (2020). Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers. *Future Generation Computer Systems*, 106, 92–104. <https://doi.org/10.1016/j.future.2020.01.005>
- Satuan Tugas Penanganan COVID-19. (2021). *Data COVID-19 di Indonesia*.

Covid19.Go.Id. <https://covid19.go.id/>

- Sitorus, I. (2020). *Support Vector Machine (SVM) and Kernels Trick*. Medium.Com. <https://medium.com/analytics-vidhya/introduction-to-svm-and-kernel-trick-part-1-theory-d990e2872ace>
- Suchetha, N. K., Nikhil, A., & Hrudya, P. (2019). Comparing the wrapper feature selection evaluators on twitter sentiment classification. *ICCIDS 2019 - 2nd International Conference on Computational Intelligence in Data Science, Proceedings*, 1–6. <https://doi.org/10.1109/ICCIDS.2019.8862033>
- Sutoyo, E., & Almaarif, A. (2020). Twitter sentiment analysis of the relocation of Indonesia's capital city. *Bulletin of Electrical Engineering and Informatics*, 9(4), 1620–1630. <https://doi.org/10.11591/eei.v9i4.2352>
- Too, J., & Mirjalili, S. (2021). A Hyper Learning Binary Dragonfly Algorithm for Feature Selection: A COVID-19 Case Study. *Knowledge-Based Systems*, 212, 106553. <https://doi.org/10.1016/j.knosys.2020.106553>
- Trivusi. (2022). *Apa itu Kernel Trick? Pengertian dan Jenis-jenis Fungsi Kernel SVM*. Trivusi.Web.Id. <https://www.trivusi.web.id/2022/04/fungsi-kernel-svm.html>
- Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415, 295–316. <https://doi.org/10.1016/j.neucom.2020.07.061>
- Yin, X., Hou, Y., Yin, J., & Li, C. (2019). A novel SVM parameter tuning method based on advanced whale optimization algorithm. *Journal of Physics: Conference Series*, 1237(2). <https://doi.org/10.1088/1742-6596/1237/2/022140>

LAMPIRAN

Lampiran 1. Kode Program Optimasi Parameter SVM Menggunakan OBWOA

```
# Error rate
def error_rate(xtrain, ytrain, x, opts):
    fold = opts['fold']
    feat = fold['feat']
    label = fold['label']
    accuracy = []

    for train_index, test_index in kfold.split(feat, label):
        X_train, X_test = feat[train_index], feat[test_index]
        y_train, y_test = label[train_index], label[test_index]
        baseline = SVC(kernel='rbf', C=x[0], gamma=x[1])
        baseline.fit(X_train, y_train)
        ypred = baseline.predict(X_test)
        accuracy.append(accuracy_score(y_test, ypred))

    error = 1 - np.mean(accuracy)
    return error

def Fun(xtrain, ytrain, x, opts):
    error = error_rate(xtrain, ytrain, x, opts)
    return error

#[2018]-"Parameter estimation of solar cells diode models by an improved
opposition-based whale optimization algorithm"
def init_position(lb, ub, N, dim):
    X = np.zeros([N, dim], dtype='float')
    for i in range(N):
        for d in range(dim):
            X[i,d] = lb[0,d] + (ub[0,d] - lb[0,d]) * rand()
    return X

def boundary(x, lb, ub):
    if x < lb:
        x = lb
    if x > ub:
        x = ub
    return x

#--- Opposition based learning (18)
def opposition_based_learning(X, lb, ub, N, dim):
    Xo = np.zeros([N, dim], dtype='float')
    for i in range(N):
        for d in range(dim):
            Xo[i,d] = lb[0,d] + ub[0,d] - X[i,d]
    return Xo

def jfs(xtrain, ytrain, opts):
    # Parameters
    ub = 10
    lb = 0
    b = 1 # constant
    N = opts['N']
    max_iter = opts['T']
    if 'b' in opts:
        b = opts['b']
    # Dimension
    dim = 2
    if np.size(lb) == 1:
        ub = ub * np.ones([1, dim], dtype='float')
        lb = lb * np.ones([1, dim], dtype='float')
```

```

# Initialize position
X = init_position(lb, ub, N, dim)

# Fitness at first iteration
fit = np.zeros([N, 1], dtype='float')
Xgb = np.zeros([1, dim], dtype='float')
fitG = float('inf')

for i in range(N):
    fit[i,0] = Fun(xtrain, ytrain, X[i,:], opts)
    if fit[i,0] < fitG:
        Xgb[0,:] = X[i,:] # posisi terbaik sejauh ini
        fitG = fit[i,0] # fitness terbaik sejauh ini

#--- Opposition based learning
Xo = opposition_based_learning(X, lb, ub, N, dim)

#--- Fitness
fit0 = np.zeros([N, 1], dtype='float')
for i in range(N):
    fit0[i,0] = Fun(xtrain, ytrain, Xo[i,:], opts)
    if fit0[i,0] < fitG:
        Xgb[0,:] = Xo[i,:]
        fitG = fit0[i,0]

#--- Merge opposite & current population, and select best N
XX = np.concatenate((X, Xo), axis=0)
FF = np.concatenate((fit, fit0), axis=0)

#--- Sort in ascending order
ind = np.argsort(FF, axis=0)
for i in range(N):
    X[i,:] = XX[ind[i,0],:]
    fit[i,0] = FF[ind[i,0]]

# Pre
curve = np.zeros([1, max_iter], dtype='float')
t = 0

curve[0,t] = fitG.copy()
print("Generation:", t + 1)
print("Best (OBWOA):", curve[0,t])
t += 1

while t < max_iter:
    # Define a, linearly decreases from 2 to 0 (14)
    a = 2 - t * (2 / max_iter)

    for i in range(N):
        # Parameter A (13)
        A = 2 * a * rand() - a
        # Parameter C (13)
        C = 2 * rand()
        # Parameter r1, random number in [0,1]
        r1 = rand()
        # Parameter l, random number in [-1,1]
        l = -1 + 2 * rand()
        # Whale position update (15)
        if r1 < 0.5:
            # {1} Encircling prey
            if abs(A) < 1:
                for d in range(dim):
                    # Compute D (12)

```

```

        Dx      = abs(C * Xgb[0,d] - X[i,d])
        # Position update (12)
        X[i,d] = Xgb[0,d] - A * Dx
        # Boundary
        X[i,d] = boundary(X[i,d], lb[0,d], ub[0,d])

    # {2} Search for prey
    elif abs(A) >= 1:
        for d in range(dim):
            # Select a random whale
            k      = np.random.randint(low = 0, high = N)
            # Compute D (16)
            Dx      = abs(C * X[k,d] - X[i,d])
            # Position update (16)
            X[i,d] = X[k,d] - A * Dx
            # Boundary
            X[i,d] = boundary(X[i,d], lb[0,d], ub[0,d])

    # {3} Bubble-net attacking
    elif r1 >= 0.5:
        for d in range(dim):
            # Distance of whale to prey (11)
            dist     = abs(X[i,d] - Xgb[0,d])
            # Position update (11)
            X[i,d] = dist * np.exp(b * l) * np.cos(2 * np.pi * l) +
Xgb[0,d]
            # Boundary
            X[i,d] = boundary(X[i,d], lb[0,d], ub[0,d])

    # Fitness
    for i in range(N):
        fit[i,0] = Fun(xtrain, ytrain, X[i,:], opts)
        if fit[i,0] < fitG:
            Xgb[0,:] = X[i,:]
            fitG     = fit[i,0]

    #--- Opposition based learning
    Xo      = opposition_based_learning(X, lb, ub, N, dim)

    #--- Fitness
    fit0    = np.zeros([N, 1], dtype='float')
    for i in range(N):
        fit0[i,0] = Fun(xtrain, ytrain, Xo[i,:], opts)
        if fit0[i,0] < fitG:
            Xgb[0,:] = Xo[i,:]
            fitG     = fit0[i,0]

    # Store result
    curve[0,t] = fitG.copy()
    print("Generation:", t + 1)
    print("Best (OBWOA):", curve[0,t])
    t += 1

    #--- Merge opposite & current population, and select best N
    XX      = np.concatenate((X, Xo), axis=0)
    FF      = np.concatenate((fit, fit0), axis=0)
    #--- Sort in ascending order
    ind     = np.argsort(FF, axis=0)
    for i in range(N):
        X[i,:] = XX[ind[i,0],:]
        fit[i,0] = FF[ind[i,0]]

# Create dictionary

```

```

obwoa_data = {'c': curve, 'Xgb' : Xgb, 'fitG' : fitG}
return obwoa_data

# parameter
N      = 10      # number of whales
T      = 100     # maximum number of iterations
b      = 1       # constant
opts = {'fold':fold, 'N':N, 'T':T, 'b':b}

# perform feature selection
parameter = jfs(X, y, opts)

# plot convergence
curve = parameter['c']
curve = curve.reshape(np.size(curve,1))
x      = np.arange(0, opts['T'], 1.0) + 1.0

fig, ax = plt.subplots()
ax.plot(x, curve, 'o-')
ax.set_xlabel('Number of Iterations')
ax.set_ylabel('Fitness')
ax.set_title('OBWOA')
ax.grid()
plt.show()

```

Lampiran 2. Kode Program Seleksi Fitur Menggunakan OBWOA

```

# error rate
def error_rate(xtrain, ytrain, x, opts):
    fold      = opts['fold']
    feat      = fold['feat']
    label     = fold['label']
    baseline = SVC(kernel='sigmoid', C=0.23757559, gamma=2.6356989)
    accuracy = cross_val_score(baseline, feat[:, x == 1], label, cv=kfold,
scoring='accuracy')
    error = 1 - np.mean(accuracy)
    return error

# Error rate & Feature size
def Fun(xtrain, ytrain, x, opts):
    # Parameters
    alpha      = 0.99
    beta       = 1 - alpha
    # Original feature size
    max_feat = len(x)
    # Number of selected features
    num_feat = np.sum(x == 1)
    # Solve if no feature selected
    if num_feat == 0:
        cost = 1
    else:
        # Get error rate
        error = error_rate(xtrain, ytrain, x, opts)
        # Objective function
        cost = alpha * error + beta * (num_feat / max_feat)
    return cost

#[2018]-"Parameter estimation of solar cells diode models by an improved
opposition-based whale optimization algorithm"
def init_position(lb, ub, N, dim):
    X = np.zeros([N, dim], dtype='float')

```

```

for i in range(N):
    for d in range(dim):
        X[i,d] = lb[0,d] + (ub[0,d] - lb[0,d]) * rand()
return X

def binary_conversion(X, thres, N, dim):
    Xbin = np.zeros([N, dim], dtype='int')
    for i in range(N):
        for d in range(dim):
            if X[i,d] > thres:
                Xbin[i,d] = 1
            else:
                Xbin[i,d] = 0
    return Xbin

def boundary(x, lb, ub):
    if x < lb:
        x = lb
    if x > ub:
        x = ub
    return x

#--- Opposition based learning (18)
def opposition_based_learning(X, lb, ub, thres, N, dim):
    Xo = np.zeros([N, dim], dtype='float')
    for i in range(N):
        for d in range(dim):
            Xo[i,d] = lb[0,d] + ub[0,d] - X[i,d]
    return Xo

def jfs(xtrain, ytrain, opts):
    # Parameters
    ub = 1
    lb = 0
    thres = 0.5
    b = 1 # constant
    N = opts['N']
    max_iter = opts['T']
    if 'b' in opts:
        b = opts['b']
    # Dimension
    dim = np.size(xtrain, 1)
    if np.size(lb) == 1:
        ub = ub * np.ones([1, dim], dtype='float')
        lb = lb * np.ones([1, dim], dtype='float')

    # Initialize position
    X = init_position(lb, ub, N, dim)

    # Binary conversion
    Xbin = binary_conversion(X, thres, N, dim)

    # Fitness at first iteration
    fit = np.zeros([N, 1], dtype='float')
    Xgb = np.zeros([1, dim], dtype='float')
    fitG = float('inf')

    for i in range(N):
        fit[i,0] = Fun(xtrain, ytrain, Xbin[i,:], opts)
        if fit[i,0] < fitG:
            Xgb[0,:] = X[i,:]
            fitG = fit[i,0]

```

```

#--- Opposition based learning
Xo = opposition_based_learning(X, lb, ub, thres, N, dim)
#--- Binary conversion
Xobin = binary_conversion(Xo, thres, N, dim)

#--- Fitness
fitO = np.zeros([N, 1], dtype='float')
for i in range(N):
    fitO[i,0] = Fun(xtrain, ytrain, Xobin[i,:], opts)
    if fitO[i,0] < fitG:
        Xgb[0,:] = Xo[i,:]
        fitG = fitO[i,0]

#--- Merge opposite & current population, and select best N
XX = np.concatenate((X, Xo), axis=0)
FF = np.concatenate((fit, fitO), axis=0)
#--- Sort in ascending order
ind = np.argsort(FF, axis=0)
for i in range(N):
    X[i,:] = XX[ind[i,0],:]
    fit[i,0] = FF[ind[i,0]]

# Pre
curve = np.zeros([1, max_iter], dtype='float')
t = 0

curve[0,t] = fitG.copy()
print("Generation:", t + 1)
print("Best (OBWOA):", curve[0,t])
t += 1

while t < max_iter:
    # Define a, linearly decreases from 2 to 0 (14)
    a = 2 - t * (2 / max_iter)

    for i in range(N):
        # Parameter A (13)
        A = 2 * a * rand() - a
        # Parameter C (13)
        C = 2 * rand()
        # Parameter r1, random number in [0,1]
        r1 = rand()
        # Parameter l, random number in [-1,1]
        l = -1 + 2 * rand()
        # Whale position update (15)
        if r1 < 0.5:
            # {1} Encircling prey
            if abs(A) < 1:
                for d in range(dim):
                    # Compute D (12)
                    Dx = abs(C * Xgb[0,d] - X[i,d])
                    # Position update (12)
                    X[i,d] = Xgb[0,d] - A * Dx
                    # Boundary
                    X[i,d] = boundary(X[i,d], lb[0,d], ub[0,d])

            # {2} Search for prey
            elif abs(A) >= 1:
                for d in range(dim):
                    # Select a random whale
                    k = np.random.randint(low = 0, high = N)
                    # Compute D (16)
                    Dx = abs(C * X[k,d] - X[i,d])

```

```

        # Position update (16)
        X[i,d] = X[k,d] - A * Dx
        # Boundary
        X[i,d] = boundary(X[i,d], lb[0,d], ub[0,d])

# {3} Bubble-net attacking
elif r1 >= 0.5:
    for d in range(dim):
        # Distance of whale to prey (11)
        dist = abs(X[i,d] - Xgb[0,d])
        # Position update (11)
        X[i,d] = dist * np.exp(b * l) * np.cos(2 * np.pi * l) +
Xgb[0,d]
        # Boundary
        X[i,d] = boundary(X[i,d], lb[0,d], ub[0,d])

# Binary conversion
Xbin = binary_conversion(X, thres, N, dim)

# Fitness
for i in range(N):
    fit[i,0] = Fun(xtrain, ytrain, Xbin[i,:], opts)
    if fit[i,0] < fitG:
        Xgb[0,:] = X[i,:]
        fitG = fit[i,0]

#--- Opposition based learning
Xo = opposition_based_learning(X, lb, ub, thres, N, dim)
#--- Binary conversion
Xobin = binary_conversion(Xo, thres, N, dim)

#--- Fitness
fit0 = np.zeros([N, 1], dtype='float')
for i in range(N):
    fit0[i,0] = Fun(xtrain, ytrain, Xobin[i,:], opts)
    if fit0[i,0] < fitG:
        Xgb[0,:] = Xo[i,:]
        fitG = fit0[i,0]

# Store result
curve[0,t] = fitG.copy()
print("Generation:", t + 1)
print("Best (OBWOA):", curve[0,t])
t += 1
#--- Merge opposite & current population, and select best N
XX = np.concatenate((X, Xo), axis=0)
FF = np.concatenate((fit, fit0), axis=0)
#--- Sort in ascending order
ind = np.argsort(FF, axis=0)
for i in range(N):
    X[i,:] = XX[ind[i,0],:]
    fit[i,0] = FF[ind[i,0]]

# Best feature subset
Gbin = binary_conversion(Xgb, thres, 1, dim)
Gbin = Gbin.reshape(dim)
pos = np.asarray(range(0, dim))
sel_index = pos[Gbin == 1]
num_feat = len(sel_index)
# Create dictionary
obwoa_data = {'sf': sel_index, 'c': curve, 'nf': num_feat}
return obwoa_data

```

```

# parameter
N     = 10     # number of whales
T     = 100    # maximum number of iterations
b     = 1      # constant
opts = {'fold':fold, 'N':N, 'T':T, 'b':b}

# perform feature selection
parameter = jfs(X, y, opts)

# plot convergence
curve = parameter['c']
curve = curve.reshape(np.size(curve,1))
x      = np.arange(0, opts['T'], 1.0) + 1.0

fig, ax = plt.subplots()
ax.plot(x, curve, 'o-')
ax.set_xlabel('Number of Iterations')
ax.set_ylabel('Fitness')
ax.set_title('OBWOA')
ax.grid()
plt.show()

```