

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu merupakan bagian penting didalam penelitian, dikarenakan penelitian terdahulu yang diperlukan untuk mengidentifikasi kekurangan, kesenjangan, dan peluang baru yang dapat dieksplorasi dari penelitian terdahulu. Hasil dari penelitian terdahulu dapat memberikan wawasan dan peluang baru untuk mendapatkan gagasan penelitian yang dilakukan yang dibandingkan dengan penelitian terdahulu. Penelitian terdahulu dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Penelitian terdahulu

Penelitian 1 (Q3)	
Nama Jurnal	Feature Selection Using New Version of V-Shaped Transfer Function for <i>Salp Swarm Algorithm</i> in Sentiment Analysis [26]
Judul Jurnal	MDPI Computation
Nama Peneliti	Dinar Ajeng Kristiyanti, Imas Sukaesih Sitanggang, Annisa, dan Sri Nurdiati.
Metode	Melakukan permodelan menggunakan K-Nearest Neighbor (KNN), Support Vector Machine, dan Naïve Bayes yang dioptimisasi dengan <i>Salp Swarm Algorithm</i> (SSA) dengan tambahan 5 transfer function yang berbeda yang akan dibandingkan dengan PSO dan ALO. Model akan dibandingkan akurasi, presisi, <i>recall</i> , skor F1, pemrosesan waktu, dan probabilitas fitur.
Hasil dan kesimpulan	SSA tanpa transfer function berhasil memberikan peningkatan pada model naïve bayes dan KNN. Peningkatan Naïve Baiyes SSA terdapat pada akurasi yang sebelumnya 0.57 menjadi 0.59. KNN SSA terdapat peningkatan di precision dari 19.5 menjadi 34.3, Recall dari 0.325 menjadi 0.425, dan F-1 dari 0.23 menjadi 0.36.
Penelitian 2 (Q3)	
Nama Jurnal	2023 7th International Conference on New Media Studies (CONMEDIA)
Judul Jurnal	Feature Selection Technique Model for Forest and Land Fire Data Sentiment Analysis: Comparison of SSA, PSO, and ALO [27]
Nama Peneliti	Dinar Ajeng Kristiyanti, Imas Sukaesih Sitanggang, Annisa, Sri Nurdiati
Metode	Melakukan optimisasi <i>feature selection</i> untuk melakukan pendeteksian opini publik di media sosial twitter dengan melakukan komparasi 3 <i>optimization</i> yaitu Particle Swarm Optimization, <i>Salp Swarm Algorithm</i> , dan Ant Lion Optimization yang akan dilakukan dengan <i>modelling</i> klasifikasi Naïve Bayes, Support Vector Machine, Dan K-Nearest Neighbor.
Hasil dan kesimpulan	Tanpa menggunakan <i>Feature selection</i> ketiga algoritma <i>classification</i> mendapat kan hasil yang bagus sebesar 89.00%–90.00%, namun setelah melakukan <i>feature selection Salp Swarm Algorithm</i> mendapatkan hasil akurasi 89.79%-

	90.02% dengan akurasi tertinggi berada di model KNN dengan hasil 90.02% dengan waktu 0.0393 detik.
Penelitian 3	
Nama Jurnal	2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)
Judul Jurnal	LSTM and GRU neural network performance comparison study : Taking Yelp Review Dataset as an Example [22]
Nama Peneliti	Shudong Yang, Xueying Yu, Ying Zhou
Metode	Membandingkan Algoritma LSTM dan GRU pada 4 dataset dengan konfigurasi data yaitu data teks panjang & kumpulan data kecil, panjang teks & kumpulan data besar, teks pendek & kumpulan data kecil, dan teks pendek & kumpulan data yang besar
Hasil dan kesimpulan	GRU menghasilkan waktu pengerjaan yang lebih cepat namun LSTM menghasilkan performa yang lebih baik dalam dataset yang lebih besar dengan kata-kata yang lebih kompleks. LSTM menghasilkan akurasi lebih tinggi di angka 75% - 79% didalam 4 dataset yang dicoba. Namun waktu pengerjaan GRU jauh lebih cepat di 129-130 detik.
Penelitian 4	
Nama Jurnal	Scientific Journal of Informatics
Judul Jurnal	A Comparative Analysis of Classification Algorithms for Cyberbullying Crime Detection: An Experimental Study of Twitter Social Media in Indonesia [23]
Nama Peneliti	Ari Muzakir , Hadi Syaputra, dan Febriyanti Panjaitan.
Metode	Melakukan Klasifikasi <i>cyberbullying</i> bahasa Indonesia menggunakan model <i>Naive Bayes</i> (NB), <i>Decision Tree</i> (DT), <i>Logistic Regression</i> (LR), dan <i>Support Vector Machine</i> (SVM) yang digabungkan dan dibandingkan dengan <i>extraction Bigram</i> , <i>Unigram</i> , <i>Trigram</i> , dan gabungan ketiganya.
Hasil dan kesimpulan	Algoritma NB Menghasilkan akurasi 66%, DT Menghasilkan akurasi 71% , LR Menghasilkan akurasi 70%, dan SVM Menghasilkan akurasi 76%. Hasil keseluruhan paling baik berada di SVM dengan Akurasi 76% presisi 78%.
Penelitian 5	
Nama Jurnal	Ingénierie des Systèmes d'Information
Judul Jurnal	Hybrid <i>Deep learning</i> Approach and Word2Vec Feature Expansion for <i>Cyberbullying</i> Detection on Indonesian Twitter [24]
Nama Peneliti	Irfan Ahmad Asqolani, Erwin Budi Setiawan
Metode	Melakukan klasifikasi <i>cyberbullying</i> dengan menggunakan algoritma <i>deeplearning</i> LSTM, CNN, CNN-LSTM.. LSTM-CNN yang dioptimisasi dengan word 2 vec dan TF-IDF
Hasil dan kesimpulan	Hasil paling baik dari keempat skenario yang dilakukan menghasilkan model LSTM-CNN dengan word 2vec menghasilkan akurasi sebesar 79,48% namun hasil ini tidak jauh berbeda dari <i>baseline model</i> yang menghasilkan akurasi 79%
Penelitian 6	
Nama Jurnal	IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS
Judul Jurnal	Hateful Sentiment Detection in Real-Time Tweets: An LSTM-Based Comparative Approach [28]
Nama Peneliti	Sanjiban Sekhar Roy, Member, IEEE, Akash Roy, Pijush Samui, Mostafa Gandomi, and Amir H. Gandomi
Metode	Melakukan komparasi algoritma LSTM, Naïve Bayes (NB), Logistic Regression (LR), Support Vector Machine (SVM), XGBoost (XGB), Random Forest (RF), K-Nearest Neighbor (k-NN), Artificial Neural Network (ANN), dan Bidirectional Encoder Representations from Transformers (BERT) dalam melakukan deteksi hate speech pada data real time yang diambil di twitter.

Hasil dan kesimpulan	Model LSTM menunjukkan hasil yang sangat baik dalam mendeteksi ujaran kebencian dengan akurasi yang paling baik mencapai 97%. Ini menunjukkan bahwa LSTM adalah pilihan yang baik untuk mengevaluasi data tekstual yang bersifat real-time dibandingkan dengan model lain seperti NB, LR, SVM, XGB, RF, k-NN, ANN, dan BERT.
Penelitian 7	
Nama Jurnal	Adaptive <i>Particle Swarm Optimization</i> algorithm based long short-term memory networks for sentimen analysis [29]
Judul Jurnal	Journal of Intelligent & Fuzzy Systems
Nama Peneliti	J Shobana dan M. Murali
Metode	Membandingkan algoritma classifier LSTM, ANN, dan SVM. Namun, dari algoritma LSTM akan di optimisasi dengan <i>Swarm optimization</i> APSO-LSTM dan PSO-LSTM yang membantu LSTM dalam memilih bobot optimal untuk lingkungan dalam jumlah iterasi yang lebih sedikit. Setiap algoritma akan di run dengan 4 dataset yang berbeda dan dibandingkan akurasi, presisi, recall, dan F-Measure.
Hasil dan kesimpulan	Dari hasil penelitian dengan menggunakan 4 dataset yang berbeda untuk setiap algoritma menghasilkan APSO-LSTM dan PSO-LSTM dibandingkan model LSTM, SVM, dan ANN. PSO-LSTM menghasilkan akurasi sebesar 92.8% - 95.1% di 4 dataset yang berbeda ini menghasilkan PSO-LSTM lebih unggul dibandingkan algoritma <i>classifier</i> biasa tanpa optimisasi namun lebih rendah dibandingkan dengan adaptive particle <i>Swarm optimization</i> .
Penelitian 8	
Nama Jurnal	Social Network Analysis and Mining
Judul Jurnal	<i>Deep learning</i> approaches for Arabic sentimen analysis [25]
Nama Peneliti	Ammar Mohammed, Rania Kora
Metode	Membandingkan algoritma deeplearning LSTM, RCNN, dan CNN.
Hasil dan kesimpulan	Dilakukan <i>running</i> setiap algoritma untuk mendeteksi data arabic <i>social media</i> yang dilakukan dalam kurun waktu 100 epoch. ditemukan LSTM tanpa <i>data augmentation</i> mendapatkan hasil yang paling tinggi dengan akurasi sebesar 81%, recall 88.9%, presisi 86,94%, dan F-Scroe 87.24% yang lebih tinggi dibandingkan dengan CNN dan RCNN. dari hasil ini dapat juga diaugmentasi yang menghasilkan hasil yang jauh lebih tinggi lagi dibandingkan dengan lstm tanpa augmentasi.
Penelitian 9	
Nama Jurnal	Acta Palaeobotanica
Judul Jurnal	IPSD: e-repository of Permian seeds from Indian Lower Gondwana [20]
Nama Peneliti	Nilay Govind, Mrutyunjaya Sahoo, Sankar Suresh Kumar Pillai, dan Suraj Kumar Sahu
Metode	Permodelan data dilakukan menggunakan model Bidirectional Encoder Representations from Transformers (BERT), yang dibandingkan dengan tiga model lainnya: Logistic Regression (LR), Support Vector Machine (SVM), dan Long Short Term Memory (LSTM). Hasil dari setiap permodelan akan dibandingkan akurasinya.

Hasil dan kesimpulan	Model BERT menghasilkan akurasi 89%, sedangkan model LR, SVM, dan LSTM menghasilkan akurasi masing-masing 75%, 74.75%, dan 65%. Klasifikasi setiap sentimen memiliki akurasi yang berkisar antara 75.88–87.33% dengan akurasi median 79.34%, yang merupakan nilai yang cukup signifikan dalam algoritma penambahan teks.
Penelitian 10	
Nama Jurnal	JURNAL RESTI
Judul Jurnal	Identifikasi Citra Pap Smear RepoMedUNM dengan Menggunakan K-Means Clustering dan GLCM [21]
Nama Peneliti	Dwiza Riana, Sri Rahayu , Sri Hadiani, Frieyadie, Muhamad Hasan, Izni Nur Karimah, Rafly Pratama
Metode	K-Means untuk mensegmentasi citra pap smear dengan melihat fitur warna sel.
Hasil dan kesimpulan	Pada penelitian ini dibuat E-Repository untuk melakukan penyimpanan dataset yang dikumpulkan dari proses akusisi citra melalui mikroskop yang menghasilkan 6168 citra data. Ketika dilakukan pengujian data dengan menggunakan k-means didapatkan rata-rata total akurasi identifikasi 92%.

Penelitian [26] pada tahun 2023 melakukan perbandingan algoritma *feature selection* pada data sentimen *analysis* dataset maskapai penerbangan pada Twitter. Penelitian ini akan membandingkan optimisasi *Salp Swarm Algorithm Using Transfer Function* (SSA-TF), *Particle Swarm Optimization* (PSO) dan *ant lion optimizer* (ALO) dengan algoritma klasifikasi k-nearest neighbor (KNN), *support vector machine* (SVM) and *naïve Bayes* (NB). Berdasarkan hasil penelitian, SSA-V3-TF dengan KNN menghasilkan hasil yang paling baik dari akurasi 80.9%, Hasil ini lebih baik dari algoritma SSA biasa yang dengan KNN menghasilkan hasil paling rendah di akurasi sebesar 55%. Namun dibandingkan algoritma lain tanpa ada optimisasi lanjutan *transfer function* algoritma PSO dan ALO menghasilkan akurasi yang cukup baik dengan KNN-PSO berada di 80.95% dan algoritma KNN-ALO berada di 80.44%.

Penelitian [27] pada tahun 2023 melakukan pembandingan algoritma *feature selection* pada data sentimen kebakaran hutan dan tanah pada platform X. Penelitian tersebut membandingkan beberapa model *feature selection* yaitu *Salp Swarm Algorithm*, *Particle Swarm Optimization*, dan *Ant Lion Optimization* dengan klasifikasi algoritma *K-Nearest Neighbour* (KNN), *Support Vector Machine* (SVM), dan *Naive Bayes* (NB). Hasil dari penelitian menunjukkan bahwa

optimisasi algoritma KNN-SSA berhasil menghasilkan akurasi paling tinggi di 90.02% dan KNN-PSO berhasil meraih *execution time* paling cepat di 0,038 detik.

Penelitian [22] pada tahun 2020 melakukan *Natural Language Processing* sentimen analisis pada *dataset review* Yelp. Penelitian ini berfokus dalam perbandingan algoritma LSTM dan GRU dalam menangani 4 dataset yang bervariasi dari kompleksitas teks dan besar dataset. Dari hasil penelitian algoritma LSTM lebih unggul dari algoritma GRU dengan tingkat akurasi didalam 4 *dataset* berkisar dari 75% sampai dengan 79% dengan waktu 164 detik sampai dengan 171 detik. Namun, hasil algoritma gru lebih unggul dari algoritma LSTM dalam waktu pemrosesan yang hanya membutuhkan waktu 128 detik sampai dengan 130 detik. Algoritma LSTM walaupun lebih lambat tapi lebih unggul dalam performa menangani beberapa *dataset* yang memiliki spesifikasi yang berbeda-beda.

Penelitian [23] pada tahun 2022 melakukan komparasi algoritma klasifikasi dalam melakukan deteksi *cyberbullying* Indonesia di media sosial twitter. Penelitian berfokus pada pendeteksian *cyberbullying* bahasa Indonesia yang dikomparasi dari algoritma klasifikasi *Naive Bayes* (NB), *Decesion Tree* (DT), *Logistic Regression* (LR), dan *Support Vector Machine* (SVM) yang digabungkan dan dibandingkan dengan *extraction Bigram*, *Unigram*, *Trigram*, dan Penggabungan ketiganya. Hasil dari penelitian menunjukkan bahwa terdapat beberapa algoritma yang menghasilkan akurasi rendah seperti NB, DT, dan LR. Hasil algoritma klasifikasi paling baik didapatkan oleh SVM dengan hasil akurasi 76%, presisi 78%, recall 80%, dan F-Measure 82% yang menjadikan nya algoritma terbaik dalam penelitian untuk mendeteksi *cyberbullying*.

Penelitian [24] pada tahun 2023 melakukan perbandingan perbandingan algoritma *hybrid deep learning* dengan bantuan word2vec untuk melakukan deteksi *cyberbullying* Indonesia di twitter. Penelitian ini membandingkan algoritma LSTM dan CNN *baseline* dan yang digabungkan dengan word2vec serta *hybrid* model LSTM-CNN dan CNN-LSTM. Penelitian ini mencoba menggabungkan *hybrid deep-learning* LSTM-CNN dan CNN-LSTM yang dibandingkan dengan algoritma *baseline deep learning* yaitu LSTM dan CNN yang dibantu dengan word

embedding word2vec dan TF-IDF. Dari hasil model diperoleh hasil akurasi tertinggi CNN-LSTM dengan akurasi 79.48%. terdapat peningkatan sebesar 0.48% dari hasil *baseline* LSTM yang menghasilkan akurasi 79.10% yang walaupun tidak signifikan, membuktikan bahwa algoritma dapat mendeteksi *cyberbullying*.

Penelitian [28] pada tahun 2023 melakukan sentimen analisis hate speech secara realtime dari media sosial twitter beberapa *Deep learning* Model. Penelitian membandingkan algoritma Naïve Bayes (NB), Logistic Regression (LR), Support Vector Machine (SVM), XGBoost (XGB), Random Forest (RF), K-Nearest Neighbor (k-NN), Artificial Neural Network (ANN), dan Bidirectional Encoder Representations from Transformers (BERT). Hasil dari penelitian menunjukkan algoritma yang paling baik dalam melakukan klasifikasi berdasarkan akurasi pada data realtime yang diambil dari twitter berada di LSTM pada tingkat akurasi 97%.

Penelitian [29] pada tahun 2021 melakukan penelitian *sentimen analysis* dengan menggunakan algoritma *deep learning* LSTM karena dapat memahami pola data teks yang lebih kompleks namun masih memiliki kekurangan didalam akurasi. Oleh karena hal tersebut penelitian ini melakukan optimisasi parameter dengan melakukan optimisasi parameter dengan *feature selection* berbasis *Swarm* yaitu *Particle Swarm Optimization* (PSO) untuk meningkatkan efesiensi dan performa model dalam melakukan *sentiment analysis*. Model yang diusulkan merupakan model optimisasi LSTM dengan *Adaptive Particle Swarm Optimization* (APSO) dan *Particle Swarm Optimization* yang dibandingkan dengan model konvensional LSTM, ANN dan SVM. Hasil dari penelitian ini model APSO-LSTM menghasilkan akurasi 97,8%, presisi 87,28%, *recall* 78,08%, dan F- Measure 82.42% yang merupakan hasil paling baik dalam penelitian. Hasil kedua terbaik di raih oleh algoritma PSO-LSTM menghasilkan akurasi 95,1%, presisi 85%, *recall* 76,5%, dan F- Measure 80,5%.

Penelitian [25] pada tahun 2019 mengenai analisa sentimen dalam bahasa arab. Penelitian ini membandingkan algoritma LSTM, RCNN, dan CNN. Dari hasil penelitian ini algoritma LSTM mendapatkan hasil yang paling tinggi dengan akurasi 81%, *recall* 88.9%, presisi 86,94%., dan F-score 87.24%.

Penelitian [20] pada tahun 2023 adalah penelitian mengenai pembuatan E-repository untuk melakukan pengumpulan data teks dan Gambar terhadap benih genus dan spesies yang dapat diupload sendiri oleh user dan disimpan ke dalam *database dataset*. Penelitian akan dijadikan referensi untuk melakukan pembuatan *repository* dimana user dapat melakukan *upload dataset* ke dalam *database* dan ditampilkan untuk digunakan sebagai *dataset* publik namun harus melakukan *login*.

Penelitian [21] pada tahun 2022 adalah penelitian pendeteksian kanker serviks *dataset* yang digunakan merupakan *dataset* yang dikumpulkan sendiri oleh peneliti dengan akuisisi citra mikroskop dan data dibuat dapat digunakan oleh banyak pihak atau *open source*. Penelitian dijadikan referensi untuk pembuatan *e-repository* dalam melakukan pengumpulan *data* mandiri dan *dataset* dapat digunakan oleh publik tanpa harus melakukan *login*.

Berdasarkan penelitian yang sudah dilakukan sebelumnya yang ditunjukkan pada Tabel 2.1, terdapat beberapa gap penelitian yang dapat diidentifikasi. Terdapat beberapa penelitian yang menggunakan algoritma *machine learning* dan *deep learning* untuk melakukan sentimen analysis *cyberbullying* di media sosial twitter dengan bahasa Indonesia [23] [24]. Dari hasil penelitian tersebut terdapat akurasi yang cukup baik di klasifikasi dengan *machine learning* di angka 70% dan dengan *deep learning* dengan angka 79%. Terdapat penelitian [22] yang mendukung penggunaan *deep learning* seperti LSTM yang dapat meningkatkan fleksibilitas algoritma ke dalam algoritma yang besar dan kompleks. Namun, didalam beberapa kasus penelitian LSTM murni jika dibandingkan dengan beberapa algoritma *deep learning* lainnya menghasilkan akurasi yang kurang di 65%. Beberapa penelitian yang dilakukan sebelumnya, membuktikan bahwa terdapat optimisasi *feature selection* berbasis *Swarm Intelligence* dapat membantu meningkatkan akurasi algoritma dan menurunkan *execution time*. dua dari penelitian terdahulu mengkomparasi algoritma Particle Swarm Optimizaton (PSO) dan algoritma *Salp Swarm Algorithm* (SSA) didalam algoritma klasifikasi *machine learning* yang terbukti dapat meningkatkan akurasi dan menurunkan waktu eksekusi [26] [27]. Mayoritas penelitian terdahulu tidak membandingkan kinerja algoritma *Particle*

Swarm Optimizaton (PSO) dan algoritma *Salp Swarm Algorithm* (SSA) untuk mencari algoritma optimisasi terbaik pada algoritma *deep learning Long Short Term Memory* (LSTM) dalam klasifikasi *cyberbullying* berbahasa Indonesia di media sosial platform X. Untuk mengisi gap penelitian yang dilakukan sebelumnya, penelitian ini akan mengadopsi optimisasi algoritma *Swarm Intelligence* dengan menggunakan algoritma *deep learning LSTM* dan membandingkannya dengan dan tanpa optimisasi *hyperparameter* berbasis *Swarm Intelligence*, algoritma akan membandingkan 2 optimisasi yaitu *Particle Swarm Optimization* (PSO) dan *Salp Swarm Algorithm* (SSA) untuk meningkatkan kinerja LSTM dalam Klasifikasi sentimen pada data *cyberbullying*.

Model dengan akurasi terbaik akan dibangun prototipe *website* untuk dapat melakukan prediksi teks *cyberbullying*. Selain itu akan dilakukan development *E-Repository database* untuk *dataset* yang digunakan didalam penelitian pembuatan *E-repository* ini didasarkan dari dua penelitian terdahulu yang melakukan pembuatan *E-repository* untuk pengumpulan data biji-bijian [20] dan pengumpulan Gambar sel kanker [21] dikarenakan tidak ada datanya *database* untuk *dataset* yang digunakan dalam penelitian. Oleh karena itu pada penelitian ini juga akan dibuat *E-repository Cyberbullying* sebagai basis *data* teks *cyberbullying* yang akan digunakan untuk penelitian.

2.2 Teori tentang Topik Skripsi

2.2.1 Media Sosial

Media sosial merupakan *platform web* dan *mobile* yang memungkinkan setiap individu untuk terkoneksi maupun terhubung dengan orang lain dalam jaringan *virtual*, *user* dapat berbagi, membuat bersama, bertukar informasi dalam bentuk konten digital termasuk informasi, pesan, foto, dan video [30]. *Sosial media* platform yang paling banyak digunakan adalah Facebook, Youtube, Platform X, dan Instagram. Media sosial memiliki peran yang sangat penting atau vital untuk membuat dunia lebih kecil karena semua orang terkoneksi. Namun dibalik keterkoneksi ini media sosial bisa menjadi pisau

2 arah yang dapat membuat efek merugikan yang besar salah satunya dikarenakan *cyberbullying* [31].

2.2.2 Media Sosial platform X

Platform X atau yang sebelumnya dikenal sebagai twitter adalah sebuah media sosial untuk komunikasi *online* yang membentuk struktur sosial yang baru. X memiliki 1,3 miliar akun dan 336 juta pengguna aktif yang memposting sekitar 500 juta tweet per hari [32]. Fitur yang membedakan platform X dengan platform media sosial lainnya adalah tweets. tweets adalah pesan yang diposting di platform media sosial platform X yang berupa "kicauan" berbasis teks hingga 280 karakter yang dapat berisi teks, foto, video, link, "@username", dan tagar yang memungkinkan pengguna mengekspresikan apa yang mereka rasakan atau pikirkan dengan kemampuan untuk meng-*update* atau men-*tag* pengguna lain dan dapat diasosiasikan ke dalam suatu topik tertentu [33].

2.2.3 Bullying

Bullying atau perundungan merupakan sebuah bentuk tindakan dan perilaku agresif, yang melibatkan perbedaan kekuatan dari korban dan pelaku yang dilakukan secara berulang dalam suatu kurun waktu. Perundungan sendiri memiliki beberapa tipe yang berbeda seperti perundungan fisik, perundungan *verbal*, perundungan relasional, dan perundungan digital atau *cyberbullying* [34].

2.2.4 Cyberbullying

Cyberbullying dapat dijelaskan sebagai tindakan agresif atau disengaja yang dilakukan oleh seorang individu maupun kelompok dengan menggunakan media elektronik secara berulang dan secara berkala dimana korban tidak dapat dengan mudahnya untuk membela dirinya sendiri. Tindakan *cyberbullying* sendiri dapat berbasis teks (didalam email atau pesan teks), visual (foto maupun video), dan verbal (pesan suara, panggilan telepon) yang dapat dilakukan melalui berbagai media, termasuk media sosial, ponsel, dan email [35].

Cyberbullying melibatkan pengiriman teks yang melecehkan atau mengancam, memposting komentar yang menghina atau mengancam secara

online. Efek dari *cyberbullying* sendiri lebih parah karena besarnya audiens yang dapat dijangkau melalui internet, dan materi yang dapat disimpan secara online membuat korban lebih sering mengingat pengalaman yang merendahkan [36].

2.2.5 Dampak Cyberbullying

Cyberbullying memiliki dampak yang sama atau lebih besar dampak negatifnya dibandingkan dengan *bullying*, ini dikarenakan predator yang menyerang korban biasanya akan menyerang korban di dalam aspek yang tidak dapat diubah seperti kulit, warna, penampilan fisik, agama, dan etnis yang meninggalkan dampak yang lebih dalam pada korban [37].

Efek dari *cyberbullying* didalam relasi sosial atau *bullying victimization* sangat berkaitan dengan *cyberbullying* dan penelitian *cross-sectional* dan *longitudal* yang telah dilakukan sebelumnya menunjukkan bahwa individu yang memiliki maupun mengalami *bullying victimization* atau *bully* lebih rentan terhadap depresi [38]. Terlebih lagi Sjursøet menemukan bahwa *cyberbullying* memiliki hubungan lebih kuat dengan gejala kecemasan dibandingkan *bullying* biasa. Bahkan *cyberbullying* sendiri maupun digabung dengan *bullying* dapat menambahkan efek negatif dalam masalah mental kesehatan remaja [6].

Depresi sendiri dapat mempengaruhi secara negatif bagaimana seseorang merasa, berpikir, dan bertindak [39]. Remaja yang mengalami tindakan *cyberbullying* memiliki gejala depresi yang lebih parah dibandingkan bentuk *bullying* lain. Perasaan ketidakberdayaan untuk mempertahankan diri dari tindakan *cyberbullying* ini dapat meningkatkan rasa ketakutan dan tekanan emosional yang menimbulkan gejala depresi [6]. *Cyberbullying* juga dapat berdampak sangat buruk kepada kesehatan mental remaja, remaja akan cenderung menyalahgunakan alkohol dan obat untuk melawan untuk melawan perasaan dari *cyberbullying* yang dialaminya. penggunaan obat ini dan efek dari *cyberbullying* sendiri dapat mendorong remaja untuk melakukan percobaan bunuh diri maupun bunuh diri [7].

Tindakan atau upaya bunuh diri yang disebabkan oleh tindakan atau pelecehan *bullying* yang dialami di sosial media disebut dengan konsep *objective Cyberbullicide*. Istilah ini menggabungkan dua konsep, yaitu "*cyberbullying*" yang merupakan intimidasi atau pelecehan yang dilakukan secara daring, dan "*suicide*" yang merupakan bunuh diri. Para korban *Cyberbullicide* sering kali mengalami pelecehan secara terus-menerus melalui media sosial berupa ancaman, ejekan, penghinaan, penyebaran rumor palsu, atau penyerangan. Kasus-kasus yang terjadi didalam *Cyberbullicide* sering kali terjadi ketika korban *cyberbullying* merasa sangat putus asa, terisolasi, dan stres hingga memutuskan untuk mengambil nyawanya sendiri. Dalam banyak kasus, korban merasa tidak memiliki jalan keluar atau dukungan yang memadai untuk menghadapi pelecehan *online* yang mereka alami [8]. Untuk membantu mengatasi tindakan *cyberbullying* yang dapat merenggut nyawa manusia dibutuhkan bantuan dalam melakukan analisa *cyberbullying* di media sosial dengan menggunakan analisis sentimen.

2.2.6 Analisis Sentimen

Pada tingginya pertumbuhan media sosial yang drastis saat ini Blog, ulasan, *tweet*, *post*, dan diskusi didalam media sosial dapat dipindai untuk melakukan ekstraksi opini dari suatu individu, Sikap, pandangan, perasaan dan pendapat merupakan bagian yang penting dalam menganalisa perilaku seseorang disebut sebagai Sentimen [40]. Analisis Sentimen merupakan sebuah bentuk dari *computational evaluation* untuk menentukan emosi halus yang diekspresikan oleh seseorang [41].

Analisa sentimen akan mengidentifikasi dan memeriksa pendapat dari individu. Opini sendiri biasanya terdiri dari entitas dan sentimen aspek yang mewakili Polaritas, nantinya aspek sentimen akan diklasifikasikan ke dalam beberapa kategori tergantung dari tujuan dari klasifikasi sentimen [42]. Analisis Sentimen akan melakukan klasifikasi berdasarkan polaritas yang disampaikan ke dalam teks misalnya positif, negatif, *netral*, mengidentifikasi sentimen target

atau topik, mengidentifikasi pemegang opini didalam aspek topik, produk atau organisasi.

2.2.7 Analisis Sentimen platform X

platform X sentimen analysis akan melakukan untuk mengevaluasi makna tersembunyi dari tweet yang diposting di platform X. Analisis Sentimen platform X merupakan sebuah tipe penelitian yang memiliki tujuan untuk menganalisa pendapat atau sentimen dari user terhadap beberapa entitas seperti topik, peristiwa, individu, masalah, layanan, dan produk serta organisasi berdasarkan konten dari tweet yang mereka buat[43].

2.2.8 Database

Database merupakan sebuah koleksi dari data atau informasi secara terorganisir dan terstruktur yang disimpan secara digital didalam program komputer. Database dirancang untuk melakukan penyimpanan, pengelolaan, dan penyediaan data secara efisien terhadap informasi yang dapat berupa angka, teks, Gambar, suara dan berbagai tipe data lainnya. Dalam melakukan pengelolaan database dibutuhkan *Database Management System (DBMS)*. DBMS adalah suatu tampilan antarmuka database yang mengelola dan menyediakan akses yang efisien terhadap basis data untuk mekanisme penyimpanan, Pengambilan, dan perubahan data. Contoh dari DBMS yang umum digunakan adalah MySQL, Oracle Database, Microsoft SQL Server, PostgreSQL [44].

2.2.9 Repository

Repository merupakan sebuah wadah atau tempat untuk melakukan penyimpanan dan pemeliharaan ratusan data, aplikasi ataupun program dalam format digital yang dapat didistribusikan atau melalui jaringan komputer yang tersedia dan dapat diakses dari internet atau media lainnya[45].

2.2.10 Data Repository

Data repository merupakan sebuah sebuah tempat atau system dimana data dapat disimpan, dikelola, dan diatur dengan cara yang terstruktur. Dalam *data repository* data dapat disimpan dalam berbagai macam format termasuk

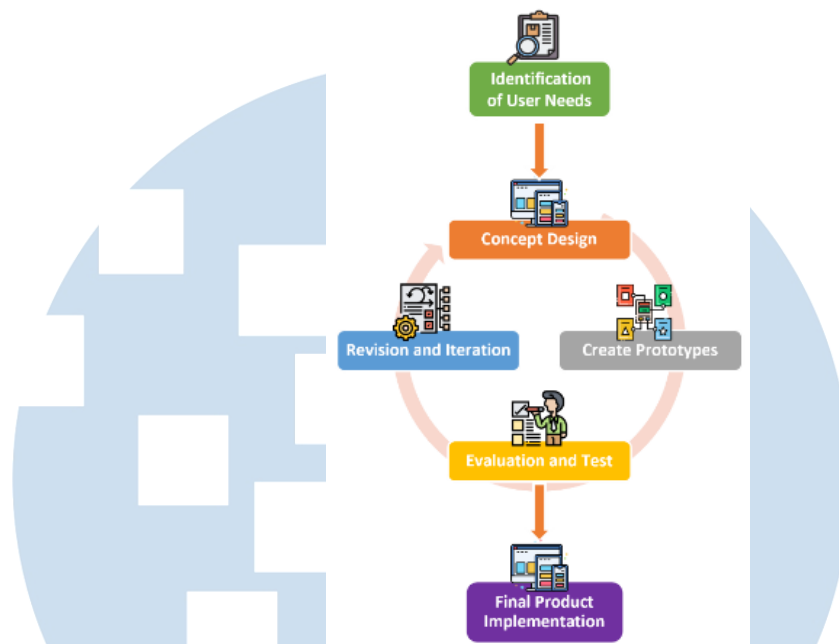
database, sistem penyimpanan berbasis file, atau dengan menggunakan platform *cloud* [46].

2.3 Teori tentang Framework dan Algoritma yang digunakan

2.3.1 Prototyping

Metode pengembangan *prototyping* adalah salah satu metode pengembangan sistem yang untuk melakukan pengembangan perangkat lunak yang melibatkan pembuatan model atau gambaran perencanaan awal dari suatu sistem perangkat lunak yang akan dikembangkan. Metode *prototyping* memungkinkan pembuat aplikasi lebih fleksibel dalam mengimplementasi perubahan selama proses pengembangan [47]. Pengembangan *prototyping* memungkinkan pengembangan aplikasi untuk dikerjakan seefisien mungkin dikarenakan proses analisis dan perancangan yang jauh lebih singkat. Secara umum, proses *Prototyping* dimulai dengan mengidentifikasi kebutuhan sistem, merancang desain awal, mengimplementasikan desain dalam bentuk *prototype*, mengevaluasi *prototype* dengan pengguna akhir, meninjau dan memperbarui *prototype* berdasarkan *Feedback*, melanjutkan pengembangan sistem, melakukan pengujian menyeluruh, dan akhirnya melakukan pemeliharaan sistem. Gambar 2.1 Menunjukkan salah satu contoh alur metode *prototyping*.





Gambar 2. 1 Metode Pengembangan *Prototyping*
Sumber: [47]




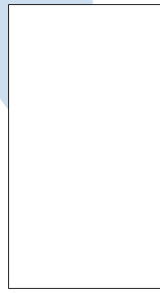
2.3.2 Unified Modelling Language

Unified Modelling Language (UML) adalah sebuah model yang digunakan untuk melakukan perancangan struktur, relasi, dan fungsi dari suatu sistem [48]. UML memungkinkan pengembang aplikasi untuk menggambarkan berbagai aspek dari sistem yang akan dibuat.

1) *Use case diagram*

Use case diagram merupakan bagian dari *UML* yang menceritakan bagaimana *system* bekerja dengan mendokumentasi *system requirement*. *Use case diagram* memungkinkan user untuk *system* yang akan dibuat tanpa harus melihat banyak detail dari *system* [49]. *Use Case diagram* akan menggambarkan interaksi yang ada diantara user dengan *system* yang akan dibuat. Dari Gambaran interaksi ini akan menjelaskan konteks dari *system* yang dibuat sehingga terlihat alur dan batasan pada sistem. Didalam *use case diagram* akan digambarkan 2 elemen yaitu user atau aktor dan *use case* yang akan menggambarkan interaksi dan relasi dari aktor ke model perilaku yang dibuat [50] [51]. Penjelasan komponen *use case diagram* dapat dilihat pada Tabel 2.2.

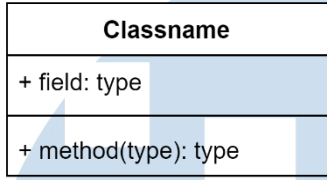

Tabel 2. 2 Komponen Use Case Diagram
 Sumber: [50]

Simbol	Nama	Keterangan
 Actor/Role	<i>Actor</i>	Pengguna sistem yang akan menggunakan sistem
 Use Case	<i>Use Case</i>	Fungsi atau aksi dari sistem yang terhubung dengan aktor
	<i>Association</i>	Garis penghubung yang menunjukkan hubungan antara aktor dengan interaksi use case
	<i>System</i>	Menandakan beberapa objek use case dan aktor yang berada pada area sistem yang sama

2) *Class diagram*

Class diagram merupakan diagram yang berorientasi pada objek, analisis, dan design yang mewakili objek dan menggambarkan hubungan antara objek dengan system yang dibuat [50] [52]. *Class diagram* akan digambarkan dengan persegi panjang (*class*) dan anak panah serta petunjuk arah yang akan menggambarkan keterkaitan diantara kelas yang ada. *Class diagram* sangat penting karena dapat membantu untuk menggambarkan secara keseluruhan dari struktur sistem serta membantu untuk memahami *software* yang sedang dibuat [53]. Penjelasan komponen *use case diagram* dapat dilihat pada Tabel 2.3.

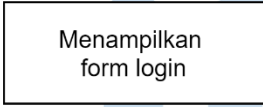




Tabel 2. 3 Komponen *class diagram*
Sumber: [50]

Simbol	Nama	Keterangan
	<i>Class</i>	Menampung objek dan atribut pada komponen pembangunan sistem.
	<i>Association</i>	Menunjukkan hubungan antara satu class dengan class yang lain

3) *Activity diagram*

Activity diagram merupakan sebuah diagram yang akan menggambarkan bagaimana *flow system* di rancang. Diagram ini akan memberikan gambaran bagaimana sebuah *flow* didalam alur sistem akan diawali, memiliki berapa opsi yang dapat terjadi, dan penyelesaian *flow* didalam sistem [50] [54].

Tabel 2. 4 Komponen *Activity Diagram*
Sumber: [50]

Simbol	Nama	Keterangan
	<i>Action/ Activity</i>	Deskripsi mengenai aktivitas atau interaksi yang dilakukan oleh suatu pihak
	<i>Initial Activity</i>	Titik awal dari sebuah interaksi atau proses
	<i>Final Activity</i>	Titik akhir dari sebuah interaksi atau proses
	<i>Decision</i>	Digunakan untuk menjelaskan kondisi yang memiliki percabangan
	<i>Line Connector</i>	Menunjukkan koneksi antara satu objek dengan objek lainnya

Simbol			Nama	Keterangan									
<table border="1"> <thead> <tr> <th colspan="3">Pool</th> </tr> <tr> <th>Lane 1</th> <th>Lane 2</th> <th>Lane 3</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>			Pool			Lane 1	Lane 2	Lane 3				Swimlane	Memberikan pembagian aktivitas diantara beberapa pihak
Pool													
Lane 1	Lane 2	Lane 3											

2.3.3 CRISP-DM

Cross-Industry Standard Process for Data Mining atau yang dikenal juga sebagai CRISP-DM merupakan sebuah metode *data mining* yang dikembangkan pada tahun 1997 sebagai panduan *framework* pengembangan *data mining* sebagai panduan bagi para pemula, komunitas, dan ahli yang dapat dimodifikasi sesuai dengan kebutuhan peneliti [55]. Didalam proses pembuatan model CRISP-DM terdapat 6 fase *data mining* [56]:

1. *Business Understanding*

Pada proses ini akan ditentukan tujuan dari penelitian dan kebutuhan apa saja yang akan diperlukan untuk penelitian. Pada tahap ini juga akan ditentukan tujuan dari melakukan *data mining* dan ditentukan juga tipe *data mining* yang dilakukan dan kriteria sukses dari *data mining*

2. *Data understanding*

Pada proses ini dilakukan proses pengumpulan dari berbagai sumber data sesuai dengan kebutuhan dan ketentuan penelitian. data yang dikumpulkan akan dicek kualitasnya dan akan ditentukan atribut dan korelasi data terhadap penelitian .

3. *Data preparation*

Pada tahap ini dilakukan pemrosesan data yang semula berasal dari *raw data* akan diolah menjadi dataset yang dapat dipakai untuk *modelling*. Pemrosesan data ini termasuk *cleaning data* dan melakukan penyesuaian data sesuai dengan model yang dibuat dan dapat disesuaikan dengan kebutuhan penelitian .

4. *Modelling*

Pada tahap ini akan dilakukan pembuatan model *data mining* yang sudah ditentukan sebelumnya akan dibuat *test case* dan permodelan sesuai dengan ketentuan dan kebutuhan penelitian sesuai dengan permasalahan dan data yang ada.

5. *Evaluation*

Pada tahap ini akan dilakukan evaluasi terhadap permodelan yang sudah dibuat terhadap tujuan dari pembuatan *data mining*. Hasil dari *modelling* yang dibuat akan dianalisa apakah sudah sesuai dengan kebutuhan penelitian yang sudah didefinisikan, seluruh proses penelitian akan *direview* di tahap ini .

6. *Deployment*

Pada tahap terakhir ini akan dilakukan integrasi atau *deployment* dari data mining ke dalam reportasi ataupun *software component* sesuai dengan kebutuhan penelitian .

2.3.4 Data Pre-Processing

Data *pre-processing* didalam platform X merupakan sebuah proses transformasi untuk mengubah data agar dapat diolah didalam algoritma yang akan dipakai dalam penelitian. Data *pre-processing* akan membantu klasifikasi untuk mengidentifikasi sentimen dan mendeteksi informasi palsu didalam teks serta mengurangi *noise* yang sangat berpengaruh dalam menentukan hasil dari algoritma klasifikasi yang digunakan. Setiap pengguna platform X memiliki gaya penulisan tanda baca yang mereka tentukan sendiri serta memiliki konten seperti *url*, *hashtag* dan *mention* yang dapat mempengaruhi penelitian [57]. Dalam melakukan pemrosesan *Data Pre-Processing* terdapat beberapa langkah umum yang harus dilakukan terutama data yang berasal dari platform X[58] [59].

1) *Remove URL, USER-Mention, Hashtag*

Kata yang tidak diinginkan atau *unicode* dikategorikan sebagai data sisa yang ada dari proses pencarian data atau data crawling, data sisa ini dapat

berkontribusi pada gangguan dalam data. Tweets yang di post oleh *user* terkadang dapat mengandung URL, emoji, hashtag atau [57].

2) *Case Folding*

Case Folding merupakan semua teks pada data yang ada dalam dataset yang sudah diambil akan diubah menjadi *lower case* atau *upper case* sesuai dengan kebutuhan pengklasifisian [60].

3) *Punctuation Removal*

Punctuation removal merupakan proses untuk melakukan penghapusan karakter-karakter yang tidak penting didalam teks seperti tanda seru, koma, titik, tanda tanya [61]

4) *Tokenization*

Tokenization merupakan tahap menentukan batas-batas didalam text diidentifikasi seperti spasi dan tanda baca. Proses ini akan memecah kalimat menjadi bagian-bagian kecil yang bermakna untuk mengidentifikasi entitas kata didalamnya [52]

5) *Normalization*

Normalisasi adalah proses mengubah teks ke dalam format standar untuk membuatnya lebih mudah diproses secara konsisten. Proses ini dapat membantu mengurangi variasi dalam teks dan membuatnya lebih mudah untuk diproses oleh algoritma pemrosesan bahasa alami dan model pembelajaran mesin. Misalnya, normalisasi bisa mengubah "Utk" menjadi "untuk", "menunda" menjadi "mengundurkan"

6) *Stop Words Removal*

Stop Word removal merupakan proses untuk menghilangkan kata-kata yang tidak penting yang tidak dapat memberikan informasi dalam indeks maupun kata-kata. kata-kata ini akan dihilangkan dengan melakukan *filtering* [62].

7) *Stemming*

Stemming merupakan sebuah proses didalam pre-processing untuk menghilangkan imbuhan awalan dan akhiran untuk mereduksi kata-kata yang

memiliki imbuhan. *Stemming* menggunakan bahasa Indonesia terdapat *library sastrawi*. *library* ini merupakan *library* pengembangan dari PHP yang menerapkan algoritma Nazier & Andriani *Library* ini akan melakukan stemming text menggunakan bahasa Indonesia menjadi kata dasar.

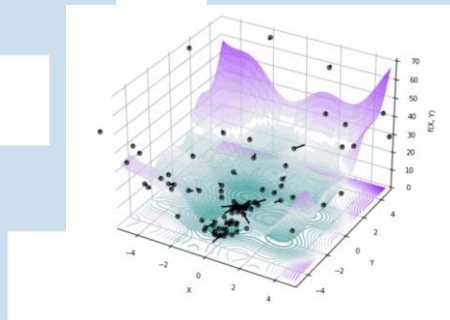
2.3.5 Swarm Algorithm

Swarm Intelligence pertama kali dikenalkan pada tahun 1989 oleh peneliti Beni dan Wang yang menjelaskan dinamika sebuah kelompok robot seluler yang dapat digabungkan menjadi sebuah bentuk kolektif yang cerdas yang memulai untuk mempelajari perilaku *Swarm* atau kawanan di luar ilmu pengetahuan alam. *Swarm Algorithm* merupakan sebuah tipe algoritma yang menggunakan konsep kecerdasan buatan yang memiliki inspirasi dari perilaku kolektif sederhana yang terdesentralisasi dan terorganisir sendiri yang mencakup interaksi sederhana dan mengikuti serangkaian aturan yang sederhana [63].

Swarm Intelligence algorithm terinspirasi dari berbagai populasi organisme biologi yang dibuat untuk meniru karakteristik atau sifat organisme ketika berinteraksi secara individu maupun didalam lingkungan untuk mencapai suatu tujuan tertentu, dengan tujuan agar interaksi ini dapat diolah menjadi *algorithm* untuk mencari sebuah solusi optimal atau mendekati optimal dengan cara heuristik dalam periode waktu yang wajar [64]. Dalam dunia hewan perilaku kawanan telah melakukan evolusi agar dapat mencapai tujuan bersama seperti melakukan pencarian makanan, membangun sarang, dan melawan musuh dengan mengandalkan karakteristik sensor motorik yang memunculkan perilaku kawanan yang berkelompok yang pola dan aturannya akan menjadi algoritma yang disebut sebagai *Swarm Intelligence* [65]. Pada saat ini *Swarm Intelligence* banyak digunakan sebagai algoritma untuk mengatasi masalah optimisasi dari algoritma *machine learning* maupun *deep learning*, beberapa algoritma yang sering dipakai didalam penelitian yang terinspirasi dari hewan seperti *particle Swarm optimization*, *salmon swarm algorithm*, *grey wolf optimization*, dan *ant lion optimization*

2.3.6 Particle Swarm Optimization

Particle Swarm Optimization (PSO) ditunjukkan pada Gambar 2.2 adalah algoritma optimisasi yang terinspirasi dari perilaku gerombolan hewan dalam alam khususnya burung atau ikan dalam melakukan pencarian makanan dengan kombinasi pengalaman sendiri dan sosial yang dibuat oleh Kennedy dan Eberhart.



Gambar 2. 2 Visualisasi PSO
Sumber: [66]

Saat melakukan pencarian makanan burung akan memperbarui posisi mereka sesuai dengan posisi terbaik menurut mereka sendiri dan posisi terbaik didalam kawanan yang menghasilkan formasi yang optimal [67]. Cara berkerja dan komputasi PSO dapat ditunjukkan pada *pseudocode* pada Tabel 2.5.

Tabel 2. 5 *Pseudocode Particle Swarm Optimization*
Sumber: [68]

Parameter masukan: ukuran populasi, jumlah iterasi, nilai minimum dan maksimum. Inisialisasi populasi partikel x_i ($i=1, 2, 3, \dots, n$) dengan nilai yang diberikan dalam batas (ub dan lb).
Selama kriteria terminasi belum terpenuhi, lakukan: Hitung nilai kecocokan setiap partikel. Perbarui parameter kognitif (c_1) menggunakan Persamaan (1).
Untuk setiap partikel: Jika partikel adalah partikel pemimpin, lakukan: Perbarui posisi partikel pemimpin menggunakan Persamaan (2). Jika tidak, lakukan: Perbarui posisi partikel pengikut menggunakan Persamaan (3).
Modifikasi posisi partikel berdasarkan batas atas dan bawah variabel.
Periksa kriteria terminasi, jika terpenuhi, keluar dari perulangan.
Kembalikan nilai F. Keluaran: Solusi terbaik global.

Dalam *Particle Swarm Optimization Algorithm* burung berkerja sebagai particle dimana keseluruhan dari kawanan burung akan berkerja sebagai particle *Swarm*. dalam melakukan pencarian setiap partikel partikel dapat dianggap sebagai individu pencarian didalam ruang pencarian N-Dimensi, kecepatan penerbangan partikel akan dapat disesuaikan dengan sejarah posisi optimal partikel dan posisi optimal historis dari sebuah populasi. Terdapat 2 properti yang ada didalam pso yaitu kecepatan yang mewakili kecepatan gerakan dan posisi yang mewakili arah gerakan. Setiap posisi dan arah gerakan akan disimpan dalam variabel *Personal Best* (Pbest) yang menyimpan posisi paling terbaik dari partikel dan *Global Best* (Gbest) yang akan menyimpan nilai kebugaran terbaik dari seluruh Pbest dan menyesuaikan posisi berdasarkan kecepatan sebelumnya, jarak Pbest, jarak Gbest dan bobot inersia yang mengatur komponen tersebut. Kecepatan akan terus ditambahkan ke posisi saat ini sehingga *particle* akan terus bergerak menuju solusi yang lebih baik. Proses ini akan diulangi terus menerus sampai kondisi berhenti tercapai, dengan partikel berkolaborasi dan beradaptasi secara dinamis, mencapai keseimbangan antara eksplorasi ruang pencarian dan eksploitasi solusi terbaik, memungkinkan PSO untuk menemukan solusi yang optimal atau mendekati optimal secara efisien untuk berbagai masalah optimisasi [69]. persamaan algoritma PSO dalam menemukan pbest dan gbest dapat dilihat pada rumus 2.1 [70].

$$\left\{ \begin{array}{l} \vec{v}_i \leftarrow \vec{v}_i + \vec{U}(0, \phi_1) \otimes (\vec{p}_i - \vec{x}_i) + \vec{U}(0, \phi_2) \otimes (\vec{p}_g - \vec{x}_i) \\ \vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i \end{array} \right\} \quad (2. 1)$$

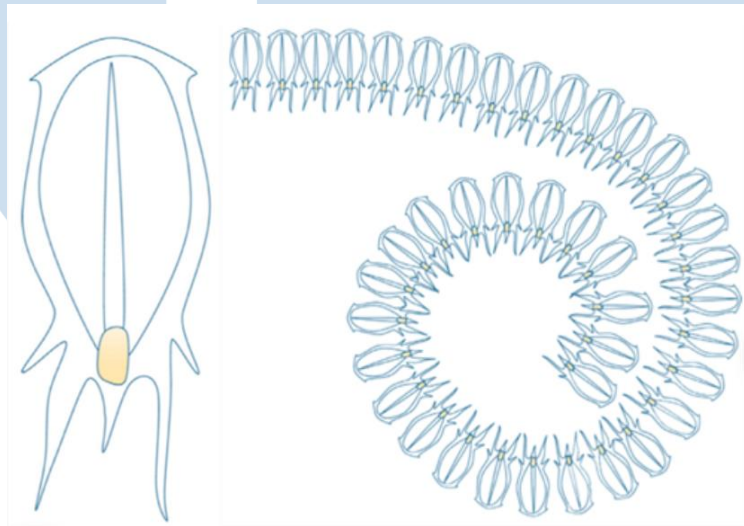
Rumus 2. 1 Rumus *Particle Swarm Optimization*

Algoritma *Particle Swarm Optimization* (PSO) yang ditampilkan pada rumus 2.1 menggunakan pembaruan posisi dan kecepatan partikel untuk mencari solusi optimal. Setiap partikel menyesuaikan kecepatannya (\vec{v}_i) berdasarkan posisi terbaik pribadi (\vec{p}_i) dan global (\vec{p}_g) dengan memasukkan faktor acak $\vec{U}(0, p)$ dan $\vec{U}(0, g)$ yang memungkinkan eksplorasi yang luas. Kemudian, posisi partikel diperbarui dengan menambahkan kecepatan baru

ke posisi saat ini, sehingga memfasilitasi pergerakan partikel ke arah yang lebih baik berdasarkan pengalaman pribadi dan kolektif dalam swarm.

2.3.7 Swarm Salp Algorithm

Salp Swarm Algorithm (SSA) merupakan algoritma yang terinspirasi dari salp dan pergerakan rantai populasi salpidae di laut didalam melakukan pencarian makanan yang diajukan oleh S. Mirjalili pada tahun 2017 [71]. Salp merupakan sebuah hewan biota laut yang memiliki badan transparan dan memiliki struktur badan seperti ubur-ubur.



Gambar 2. 3 Hewan Salp dan rantai Salp
Sumber: [26]

Algoritma salp diinspirasi oleh perilaku kelompok salp yang terhubung dari ujung ke ujung yang membentuk sebuah rantai yang bergerak secara berurutan seperti pada Gambar. Rantai salp ditunjukkan pada Gambar 2.3 memiliki seorang pemimpin yang memiliki penilaian optimal terhadap lingkungan dan sering kali berada di puncak rantai makanan, namun pemimpin secara langsung tidak akan mempengaruhi pergerakan kelompok tetapi akan mempengaruhi pergerakan salp di sebelahnya [72]. Cara bekerja komputerisasi dari *Salp Swarm Algorithm* dapat dilihat dari Tabel 2.6

Tabel 2. 6 Pseudocode Salp Swarm Algorithm Sumber: [73].

Inisialisasi populasi salp (x_i dengan $i=1, 2, 3, \dots, n$) dengan nilai batas atas (ub) dan batas bawah (lb).
--

Selama kriteria terakhir belum terpenuhi, lakukan: Hitung nilai kecocokan setiap agen pencarian (salp) menggunakan nilai F yang diperoleh dari mesin pencari tier atas.. Modifikasi nilai c1 menggunakan Persamaan (1).
Untuk setiap salp: Jika salp adalah salp pemimpin, lakukan: Perbarui posisi salp pemimpin menggunakan Persamaan (2). Jika salp adalah salp pengikut, lakukan: Perbarui posisi salp pengikut menggunakan Persamaan (3).
Modifikasi posisi salp berdasarkan batas atas dan bawah variabel
Evaluasi kriteria terminasi, jika terpenuhi, keluar dari perulangan.
Keluarkan nilai F sebagai solusi terbaik global..

SSA berkerja dengan cara menirukan perilaku rantai salp dalam melakukan pencarian makanan, dimana setiap rantai akan dibagi 2 populasi menjadi yaitu pemimpin dan pengikut. Pemimpin berada di depan rantai dan akan memperbarui posisinya berdasarkan lokasi sumber makanan (F) dan batasan atas (ub) serta batasan bawah (lb) dengan faktor acak (p1,p2,p3) yang akan menentukan arah dan besar langkah yang ditentukan untuk menyeimbangkan eksplorasi dan eksploitasi. setiap iterasi yang dilakukan, pemimpin akan mendorong pengikut untuk memperbarui posisi mereka dengan mengambil rata-rata posisi antara mereka sendiri dan salp yang berada di samping mereka. Proses ini akan terus berlangsung dengan salp pengikut secara terus menerus yang akan menyesuaikan posisi mereka berdasarkan pemimpin untuk melakukan pencarian posisi optimum secara global dalam ruang pencarian n-dimensi [74]. Berikut merupakan rumus 2.2 persamaan Salp *Swarm Algorithm* [73].

$$x_j^1 = \begin{cases} F_j + c_1 + ((ub_j - lb_j) c_2 + lb_j) c_3 \geq 0 \\ F_j - c_1 + ((ub_j - lb_j) c_2 + lb_j) c_3 < 0 \end{cases} \quad (2. 2)$$

$$c_1 = 2e - \left(\frac{4L}{L}\right)^2$$

$$x_j^i = \frac{1}{2}at^2 + v_0t$$

$$x_j^i = \frac{1}{2}(x_j^i + X_j^{i-1})$$

Rumus 2. 2 Salp *Swarm Algorithm*

Didalam rumus 2.2 algoritma SSA terdapat beberapa parameter perlu diinisialisasi yaitu Jumlah salp (S), jumlah iterasi (A), posisi salp optimal (Z^*), dan nilai kecocokan terbaik ($f(Z^*)$). Awalnya, populasi S salp dihasilkan secara acak, dan kecocokan setiap salp dievaluasi. Jumlah iterasi (a) diatur menjadi nol, dan $r1$ diperbarui sesuai dengan Persamaan (2). Untuk setiap salp, jika itu adalah salp pemimpin ($m == 1$), posisinya diperbarui menggunakan Persamaan (1). jika itu adalah salp pengikut, posisinya diperbarui menggunakan Persamaan (4). Setelah pembaruan ini, kecocokan dari setiap salp dihitung ulang.

2.3.8 Algoritma LSTM

Long Short-Term Memory Network (LSTM) adalah algoritma yang diusulkan pada tahun 1997 oleh Hochreiter and Schmidhuber untuk mengatasi masalah kehilangan gradien dan ledakan gradien dengan memperkenalkan mekanisme fungsi gerbang atau *gated function* [75]. LSTM memungkinkan dilakukan penyimpanan beberapa ketergantungan temporal dengan mengubah vektor keadaan sel yang disebarkan untuk menangkap ketergantungan long-term dependencies [76]. LSTM Lebih mementingkan prioritas informasi dibandingkan dengan waktu, hal ini membuat LSTM cocok dalam memprediksi data berbasis *time-series*, konteks temporal ataupun urutan teks karena kemampuannya untuk menyimpan ingatan dari input sebelumnya. Didalam *sentimen analysis* konteks merupakan hal yang sangat penting karena kata-kata dalam sebuah kalimat sering kali memiliki makna yang berbeda tergantung dari kata sebelumnya yang berada di kalimat. LSTM dapat mengatasi masalah ini dengan menggunakan perubahan sentimen dan teks yang lebih kompleks dan panjang. Salah satu optimisasi yang dapat dilakukan didalam algoritma LSTM adalah *objective function*. *Objective function* akan melakukan eliminasi prediksi yang salah pada *hyperparameter* sehingga hasil yang didapatkan adalah hasil prediksi yang terbaik. LSTM yang dioptimisasi dengan *objective function* dapat membantu mengoptimisasi kinerja LSTM dengan melakukan prediksi rekomendasi jumlah unit LSTM, dengan

optimisasi ini diharapkan model LSTM dapat meningkatkan akurasi dan mengurangi *loss*. Rumus LSTM dapat dilihat pada persamaan (2.3) [77][78].

$$\begin{aligned}
 & \bullet f_t = \sigma(W_f \cdot [x_t, h_{t-1}] + b_f) \\
 & \bullet i_t = \sigma(W_i \cdot [x_t, h_{t-1}] + b_i) \\
 & \bullet \tilde{c}_t = \tanh(W_c \cdot [x_t, h_{t-1}] + b_c) \\
 & \bullet C_t = f_i \cdot C_{t-1} + i_t \cdot \tilde{C}_t \\
 & \bullet O_t = \sigma(W_o \cdot [x_t, h_{t-1}] + b_o) \\
 & \bullet h_t = O_t \cdot \tanh(c_t)
 \end{aligned} \tag{2.3}$$

Rumus 2.3 Long Short Term Memory

- *Forget Gate* f_t akan menentukan seberapa banyak informasi dari sel memori sebelumnya yang akan dilupakan atau dipertahankan. *Forget gate* menentukan fungsi *sigmoid* untuk menentukan nilai antara 0 dan 1 dari setiap elemen dalam sel memori.
- *Input Gate* i_t adalah komponen yang mengatur seberapa banyak informasi baru yang masuk ke dalam memori. *Input gate* akan menggunakan *sigmoid* untuk menentukan seberapa banyak informasi dari input yang akan disimpan ke dalam memori.
- *Candidate cell state* C_t Adalah komponen yang akan menghasilkan nilai kandidat baru yang akan ditambahkan ke dalam sel memori. *Candidate cell* menggunakan fungsi *tanh* untuk menghasilkan vektor kandidat berdasarkan input saat ini dan dari sel sebelumnya.
- *Update cell state* C_t adalah tahap sel memori akan secara aktif diperbarui dengan informasi baru yang relevan dari vektor kandidat yang dihasilkan *candidate cell*. Proses ini melibatkan 3 langkah: pertama, informasi dari sel memori lama akan di nontaktifkan dan dihapus sebagian oleh *forget gate*, kedua informasi baru dari *candidate cell* diperitmbangkan dan diintegrasikan ke dalam sel dengan *input gate*, terakhir sel memori akan diperbarui dan digabungkan dengan informasi lam yang telah dibuang dengan informasi baru.

- *Output gate O_t* adalah pengontrol banyaknya informasi yang disimpan dan dipindahkan ke sel output. *Output gate* memungkinkan LSTM untuk memilih informasi penting untuk dipertahankan dan dipindahkan ke langkah berikutnya dan mengabaikan informasi yang tidak relevan.
- *Output h_t* dari lstm ada representasi dari informasi yang terkandung di dalam sel lstm. *Output* sering kali diartikan sebagai representasi fitur atau dilanjutkan ke lapisan berikutnya. *Output* dihasilkan dengan mempropagasi input melalui fungsi aktivasi yang sesuai seperti fungsi tanh atau sigmoid, tergantung pada kebutuhan tugas. Hal ini memungkinkan model memahami dan mengambil keputusan berdasarkan pola yang terkandung dalam data pada langkah waktu tersebut.

2.3.9 Statistical Test

Statistical test adalah sebuah *tools* matematika didalam *machine learning* yang digunakan untuk menganalisa data kuantitatif yang dihasilkan dalam suatu studi penelitian [79]. *Statistical test* merupakan suatu metode untuk menentukan apakah suatu pernyataan, seperti hipotesis mengenai karakteristik kuantitatif dari populasi dapat diterima atau ditolak. Proses ini melibatkan pengambilan sampel acak dari sebuah populasi yang relevan dan melakukan perhitungan statistik dari elemen-elemennya. Tujuan dari *Statistical test* dalam model *deep learning* adalah untuk membuat keputusan berdasarkan hasil dari sampel dan hasil yang dilakukan. Dalam penelitian ini, *statistical test* digunakan untuk melakukan validasi model untuk membuktikan terdapat perbedaan antara model-model yang digunakan berdasarkan parameter seperti akurasi, *loss*, dan waktu eksekusi model. Pada penelitian ini terdapat dua *statistical test* yang digunakan pada penelitian ini:

1) *Shapiro-Wilk Test*

Shapiro-Wilk Test adalah salah satu metode *statistical test* yang paling populer untuk melakukan pengujian normalitas dari data [80]. Shapiro test akan melakukan pengujian berdasarkan nilai *value* dari *expected value*

yang normal standar dan dari nilai rata-rata dari sampel [81]. Persamaan shapiro-wilk test dapat dilihat pada persamaan [54] (2.4)

$$W = \frac{(\sum_{i=1}^n a_i x_{(i)})^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (2.4)$$

Rumus 2. 4 Shapiro Wilk Test

Rumus 2.4 merupakan penjelasan dari persamaan shapiro

- W adalah nilai statistik W Shapiro-Wilk nilai mendekati 1 menunjukkan distribusi yang lebih normal
- N adalah jumlah pengamatan dalam sampel.
- a_i adalah koefisien yang ditentukan oleh Tabel distribusi berdasarkan urutan dan jumlah data n untuk memberikan bobot pengamatan.
- x_i adalah pengamatan individu dalam sampel data.
- \bar{x} adalah rata-rata sampel.

2) *Analysis of variance (Anova) Test*

ANOVA adalah sebuah metode uji statistik yang digunakan untuk membandingkan rata-rata tiga kelompok atau lebih yang berbeda untuk menentukan apakah terdapat signifikansi antara kelompok . Anova biasanya digunakan ketika ketika didalam dataset terdapat satu variabel dependen dan satu atau lebih variabel independen .

$$F = \frac{MS_{between}}{MS_{within}} \quad (2.5)$$

Rumus 2. 5 ANOVA

Rumus 2.5 adalah penjelasan dari ANOVA

- D adalah nilai uji F
- MSB adalah Mean square between dihitung dengan membagi jumlah variabilitas antar kelompok dengan jumlah derajat kebebasan antar kelompok.

- Mean square within dihitung dengan membagi jumlah variabilitas dalam kelompok dengan jumlah derajat kebebasan dalam kelompok.

2.3.10 User Acceptance Testing

User Acceptance Testing (UAT) merupakan tahap terakhir yang dilakukan ketika aplikasi sudah selesai dilakukan development dengan tujuan sebagai evaluasi akhir dan pengecekan system apakah sudah sesuai kebutuhan user [82]. Selama UAT, aplikasi menjalani evaluasi dunia nyata oleh pengguna yang menggunakan aplikasi dari segi fungsionalitas dan kinerja. Tahap ini penting karena pengguna memanfaatkan perangkat lunak itu sendiri sebagai alat utama untuk komunikasi dan dialog, memastikan bahwa kebutuhan pengguna dipahami secara tepat dan diterapkan secara akurat dalam aplikasi. Dengan mengintegrasikan pendekatan ini UAT dapat meminimalkan risiko miskomunikasi, menyelaraskan fungsionalitas perangkat lunak dengan kebutuhan pengguna [83].

2.3.11 Blackbox Texting

Blackbox testing atau yang disebut juga *behavioral testing* adalah metode pengetesan yang hanya berfokus pada *input* dan *output* dari sebuah *software* yang didasarkan dari kebutuhan *software* dan spesifikasi *software* [84]. *Blackbox testing* bertujuan untuk melakukan identifikasi kesalahan atau *bug* didalam *system* yang dapat mempengaruhi performa aplikasi dan kenyamanan pengguna tanpa memperhatikan kode program yang diimplementasikan.

2.3.12 VADER

Valance Aware Dictionary and Sentiment Reasoner (VADER) merupakan sebuah modul *rule based* Analisis Sentimen *tool* yang menggunakan sentimen lexicon dan bersifat *open source* dengan lisensi dari MIT [85]. Vader dapat membantu untuk menghitung *sentiment score* yang dibagi menjadi 3 yaitu positif, negatif, dan netral. Vader menggunakan *Natural Language Processing Tool Kit* (NLTK) untuk memberikan *sentiment score* berdasarkan kata-kata yang

ada dan bergantung pada kamus *sentiment related works* untuk melakukan klasifikasi sentimen [86].

Vader memiliki beberapa keunggulan yaitu dapat berkerja untuk text dari sosial media, menggunakan *standart lexicon* untuk melakukan generalisasi berbasis valence dengan kurator manusia sehingga tidak memerlukan training data, mendukung *emoji* untuk melakukan klasifikasi sentimen, merupakan algoritma yang relatif cepat, memiliki *speed-performance tradeoff* yang minimal [85].

2.3.13 Google Cloud Platform Translation API

Google Cloud Translation API adalah layanan yang disediakan oleh *Google Cloud Platform* yang memungkinkan user untuk melakukan penerjemahan teks dari satu bahasa ke bahasa lain secara otomatis dengan bantuan pemrosesan bahasa alami. *Google Cloud API* memungkinkan user untuk melakukan proses translasi bahasa dalam jumlah yang besar dan dapat menerjemahkannya ke dalam kualitas yang lebih tinggi. *Google Cloud Translation API* memungkinkan user untuk melakukan translasi lebih dari 1000 bahasa yang dapat diterjemahkan dalam berbagai konteks dan lingkungan multi bahasa yang efisien dalam melakukan translasi teks dalam skala besar.

2.4 Tools

2.4.1 Python

Python merupakan sebuah bahasa pemrograman interpretatif dan multiguna yang dibuat oleh Guido van Rossum pada tahun 1990. Python merupakan bahasa pemrograman yang gratis yang dapat digunakan untuk keperluan komersial. Python menggabungkan kapabilitas, kemampuan dengan *syntax code* yang jelas dan dilengkapi fungsionalitas dari library yang komprehensif [87]. Python adalah salah satu bahasa pemrograman yang banyak digunakan untuk melakukan *machine learning* dan *deep learning* dikarenakan mudah dipelajari namun dapat memiliki kekuatan sistem bahasa pemrograman tingkat tinggi apabila diperlukan serta didukung oleh berbagai macam *library*

yang meringankan beban komputasi dan memudahkan penggunaan *machine learning* kepada user [88].

2.4.2 JupyterLab

Jupyterlab merupakan sebuah web based application yang di develop oleh jupyter *enviroment*, sebuah proyek nirlaba yang membuat software *open source* berbasis website yang dapat digunakan untuk melakukan komputasi pemrograman berbagai bahasa [89]. Jupyterlab memungkinkan user untuk melakukan interaksi visualisasi data didalam *website* termasuk melakukan pembuatan notebook, pengetikan code, dan pengolahan data yang memungkinkan dilakukannya *modeling machine learning* maupun *deep learning*.

2.4.3 Visual Studio Code

Visual studio code atau yang dikenal juga sebagai VScode merupakan aplikasi tidak berbayar yang bersifat *open source* yang dikembangkan oleh microsoft untuk melakukan pemrograman dalam berbagai bahasa pemrograman seperti C, C++, C#, java, java script, php, python, dan berbagai macam bahasa pemrograman lainnya. Visual studio code menawarkan berbagai fitur-fitur yang dapat memudahkan pengerjaan program ataupun aplikasi seperti *highlight* pada kode *syntax*, *Intellisense* (pemenuhan kode cerdas yang direkomendasikan sesuai dengan input user), *debugging*, *terminal* [90].

2.4.4 HTML

HTML (Hypertext Markup Language) adalah Bahasa markup yang terstandarisasi yang paling sering digunakan oleh *web programmer* untuk merancang aplikasi web. HTML digunakan untuk mendefinisikan struktur dasar dari sebuah halaman web seperti *layout*, teks, Gambar, *hyperlink*, dan berbagai elemen lainnya yang dapat terkandung didalam sebuah website. HTML menggunakan berbagai tag dan elemen yang dapat ditempatkan didalam *document* untuk menentukan bagaimana konten dan format dari sebuah halaman *website* akan ditampilkan [91].

2.4.5 JavaScript

JavaScript adalah bahasa pemrograman yang serbaguna yang utamanya digunakan untuk pengembangan web, memungkinkan pengembang untuk menambahkan elemen interaktif ke situs web. Ini adalah bahasa tingkat tinggi yang diinterpretasikan yang mendukung gaya pemrograman berorientasi objek, imperatif, dan fungsional. JavaScript umumnya digunakan untuk *scripting* di sisi klien dalam browser web untuk membuat pengalaman pengguna yang dinamis dan interaktif. Ini juga dapat digunakan di sisi server melalui platform seperti Node.js, memperluas kemampuannya untuk membangun aplikasi web full-stack. Populeritas dan fleksibilitas JavaScript telah menyebabkan adopsi dalam berbagai lingkungan pengembangan di luar browser web, termasuk pengembangan aplikasi seluler dan pemrograman di sisi server [92].

2.4.6 MySQL

MySQL merupakan sebuah sistem manajemen database *relational database management system* (RDBMS) yang berbasis *client side* yang dibuat oracle dan bersifat *open source database*. MySQL dikenal dengan efisiensi dalam handle data dalam jumlah banyak dan memiliki performa yang bagus dalam menjalankan database. MySQL juga mensupport banyak bahasa pemrograman sekaligus seperti PHP, Java, Python, dan C++ yang dapat dipakai didalam banyak aplikasi termasuk aplikasi berbasis website [93].

2.4.7 Flask

Flask adalah *framework* pengembangan web berbasis python yang ringan dan fleksibel serta dirancang khusus untuk memudahkan pembuatan aplikasi berbasis *website* dan API. Flask memudahkan user dalam melakukan penggunaan dan pendekatan yang minimalis dikarenakan dalam memulai proyek tidak diperlukan konfigurasi yang terlalu rumit dan berlebihan yang membuat development aplikasi bisa menjadi lebih cepat [94]. Framework flask menawarkan fleksibilitas yang tinggi dan mendukung pengembang aplikasi web dengan memanfaatkan ekstension yang telah disediakan.

Kelebihan dari flask adalah flask menggunakan arsitektur yang ringan dan modular, hal ini membuat flask menjadi pilihan yang ideal untuk pengembangan web yang dibutuhkan secara cepat [95]. Selain itu kode yang digunakan didalam flask lebih mudah dipahami dan memiliki berbagai macam dokumentasi dan komunitas yang aktif untuk membantu mengembangkan website. Dokumentasi flask dapat memberikan panduan yang jelas tentang penggunaan framework yang didukung dengan komunitas yang aktif memberikan dukungan dan bantuan didalam forum diskusi dan proyek repositori seperti github [96].

2.4.8 React JS

React JS atau dikenal juga dengan React Java Script adalah sebuah library open source yang dikembangkan oleh Jordan Walke untuk melakukan pembuatan *user Interface* pada aplikasi website pada suatu page [97]. React JS memudahkan *user* untuk melakukan pengelolaan kode dengan melakukan pendekatan berbasis komponen dan pengelolaan kode untuk meningkatkan efisiensi pengembangan.

2.4.9 Express

Express adalah *framework* aplikasi web yang berbasis node js yang digunakan untuk melakukan pembuatan aplikasi web dan api. Express menyediakan berbagai *tools* untuk mengatur *routing* http, membuat dan menangani *request* dan *response* dari HTTP atau api. Express menyediakan berbagai macam modul library dan extension untuk membantu pengembangan aplikasi yang lebih kompleks yang didukung juga oleh komunitas yang menjadikan express adalah satu *backend* untuk membantu aplikasi website yang efisien

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A