

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Obesitas

Obesitas didefinisikan sebagai akumulasi lemak abnormal atau berlebihan yang beresiko bagi kesehatan. Obesitas pada usia produktif terjadi karena tidak seimbang antara asupan kalori dengan aktivitas fisik yang dilakukan sehari-hari. Obesitas merupakan faktor resiko terjadinya penyakit kronis seperti diabetes, hipertensi, jantung dan kanker dan lain-lain.(Agustina et al., 2023)

Diagnosis kelebihan berat badan dan obesitas ditegakan dengan mengukur tinggi badan dan berat badan seseorang. Selain masalah fisik, obesitas juga berdampak pada kesehatan mental dan kesejahteraan psikologis seseorang. Individu yang mengalami obesitas mungkin mengalami stigma sosial, depresi, dan rendahnya harga diri. Dengan memahami kompleksitas obesitas dan dampaknya yang luas, penting bagi individu, masyarakat, dan penyedia layanan kesehatan untuk bekerja sama dalam upaya pencegahan, pengelolaan, dan penanganan obesitas.

2.1.1.1 Faktor Penyebab Obesitas

Obesitas adalah kondisi kompleks yang disebabkan oleh berbagai faktor, termasuk pola makan yang buruk (tinggi kalori, rendah serat), kurangnya aktivitas fisik hingga faktor lingkungan (polusi udara, zat kimia dalam makanan). Pengelolaan obesitas membutuhkan pendekatan holistik yang memperhitungkan semua faktor yang relevan.

2.1.1.2 Pencegahan Obesitas

Beberapa alternatif untuk mencegah obesitas dan hidup lebih sehat dengan menjaga pola makan dan meningkatkan aktivitas fisik. Program pendukung diet seimbang dan aktivitas fisik yang dapat diterapkan pada remaja meliputi: pendidikan kesehatan melalui pesan teks singkat dan booklet, dan video audio visual, pemantauan berat badan melalui aplikasi diet. (Ida Niara et al., n.d.)

Berolahraga di tempat gym memiliki dampak positif dalam proses penurunan berat badan, sebagaimana dijelaskan oleh ahli kesehatan Indonesia, Profesor Amin Subagio, "Melalui latihan di gym, massa otot dapat ditingkatkan, sehingga tubuh mampu membakar lebih banyak kalori, bahkan saat beristirahat." Pernyataan ini didukung oleh penelitian yang dilakukan oleh Dr. Fitriani Wibowo dari Universitas Gadjah Mada, yang menemukan bahwa "Otot memerlukan energi lebih banyak daripada lemak, sehingga peningkatan massa otot dapat meningkatkan metabolisme tubuh secara keseluruhan." Dengan demikian, rutin berolahraga di gym tidak hanya membantu mencapai berat badan yang sehat, tetapi juga mempromosikan pemeliharannya dalam jangka panjang, sebagaimana diungkapkan oleh ahli nutrisi Indonesia, Dr. Putri Santoso.

Selain manfaat fisik, berolahraga di gym juga memiliki dampak positif pada kesehatan mental. Aktivitas fisik telah terbukti dapat mengurangi stres, meningkatkan mood, dan mengurangi risiko depresi. Dengan merasa lebih baik secara emosional, seseorang cenderung memiliki motivasi yang lebih besar untuk menjaga gaya hidup sehat secara keseluruhan, termasuk pola makan yang seimbang. Oleh karena itu, gym bukan hanya tempat untuk membentuk tubuh, tetapi juga tempat untuk merawat kesehatan secara menyeluruh.

2.1.2 Machine Learning

Machine learning adalah bidang ilmu komputer yang berfokus pada pengembangan algoritma yang memungkinkan sistem untuk belajar dari data. Ada beberapa jenis machine learning, namun secara umum, ada 2 kategori utama yaitu *Supervised Learning* (Pembelajaran Terawasi) dan *Unsupervised Learning* (Pembelajaran Tanpa Pengawasan).

Dalam *supervised learning*, model ML belajar dari contoh data yang berpasangan dengan label. Artinya, model diberikan input dan output yang diharapkan, dan tujuannya adalah untuk belajar hubungan antara input dan output. Terdapat beberapa jenis algoritma *Supervised* yaitu Klasifikasi, Regresi, *Forecasting* atau *Time Series*.

2.1.2.1 Metode Klasifikasi

Metode klasifikasi dalam *machine learning* memungkinkan untuk memprediksi kategori atau label dari data baru berdasarkan pola yang ditemukan dalam data latihan. Metode ini merujuk pada teknik atau algoritma yang digunakan untuk memprediksi kategori atau label dari data baru berdasarkan pola yang ditemukan dalam data latihan yang telah diberi label. Tujuannya adalah untuk mengelompokkan atau mengklasifikasikan data ke dalam kategori atau kelas yang sesuai berdasarkan fitur atau atribut yang ada.

Metode Klasifikasi terdapat beberapa algoritma diantaranya *Decision Tree*, *Naive Bayes Classifier*, *K-Nearest Neighbors* (KNN), *Support Vector Machines* (SVM) dan *Random Forest*.

2.1.2.2 *K-Nearest Neighbors*

K-Nearest Neighbor atau *KNN* merupakan sebuah algoritma klasifikasi dalam *machine learning* yang menggunakan data pembelajaran sebagai acuan untuk menentukan kategori atau kelas dari objek baru. Algoritma ini menghitung jarak antara objek baru dan setiap data pembelajaran untuk menentukan kelas yang paling cocok. Dengan demikian, *KNN* dapat mengklasifikasikan objek baru berdasarkan atribut dan sampel pembelajaran yang telah ada. *K-Nearest Neighbor* adalah algoritma yang biasanya digunakan untuk mengklasifikasikan objek baru dengan mempertimbangkan mayoritas kategori pada *KNN* dari data pembelajaran yang telah ada. Kelas dengan frekuensi terbanyak akan menjadi hasil klasifikasi dari metode *KNN*.(Fuansah et al., n.d.)

1. **Cara Kerja KNN** : Menggunakan data pelatihan secara langsung dan membuat prediksi berdasarkan kedekatan (jarak) dengan tetangga terdekat
2. **Kelebihan KNN** :
 1. Sederhana dan mudah diimplementasikan.
 2. Tidak memerlukan asumsi tertentu tentang distribusi data.
 3. Cocok untuk dataset dengan pola yang kompleks dan non-linear.
 4. Dapat digunakan untuk klasifikasi dan regresi.
3. **Kekurangan KNN** :
 1. Sensitif terhadap noise dan fitur-fitur yang tidak relevan.
 2. Membutuhkan penyimpanan yang besar karena harus menyimpan seluruh dataset.
 3. Komputasi yang mahal secara komputasional, terutama pada dataset besar.

2.1.2.3 Support Vector Machines

Algoritma *Support Vector Machine* (SVM) sendiri ialah algoritma yang bertujuan untuk menemukan hyperplane maksimal, hyperplane adalah suatu fungsi yang dapat memisahkan antara dua kelas. Pada prosesnya *SVM* akan memaksimalkan margin atau jarak antara pola pelatihan dan batas keputusan (Abdusyukur, 2023)

SVM juga dapat digunakan untuk masalah klasifikasi dengan lebih dari dua kelas. Dapat menggunakan pendekatan *One-vs-All* atau *One-vs-One* untuk mengadaptasikannya. Pendekatan *One-vs-All* melibatkan melatih model SVM untuk setiap kelas dengan memperlakukan kelas tersebut sebagai positif dan yang lainnya sebagai negatif. Sedangkan pendekatan *One-vs-One* melibatkan melatih model *SVM* untuk setiap pasangan kelas yang mungkin. Dalam kedua pendekatan ini, prediksi akhir diperoleh dari model-model individu yang diperoleh.

1. **Cara Kerja SVM** : Membuat model yang memisahkan kelas dengan hyperplane dalam ruang fitur, menggunakan data pelatihan untuk menentukan hyperplane terbaik.

2. Kelebihan *SVM* :

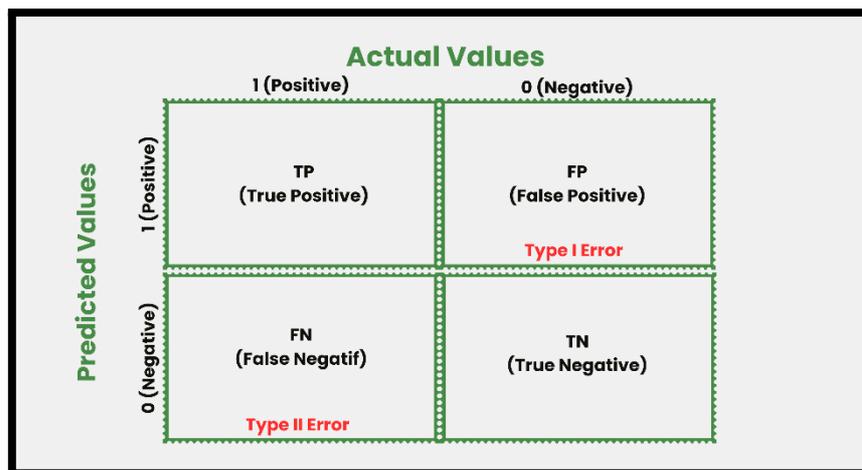
1. Efektif dalam ruang fitur yang memiliki dimensi tinggi.
2. Tidak sensitif terhadap dimensi fitur yang tinggi (*curse of dimensionality*).
3. Dapat menangani dataset dengan jumlah fitur yang lebih besar daripada jumlah sampel.
4. Cocok untuk dataset yang memiliki pola yang kompleks dan non-linear dengan menggunakan kernel.

3. Kekurangan SVM :

1. Kompleksitas komputasi yang tinggi terutama pada dataset besar.
2. Memerlukan penyesuaian parameter yang cermat, seperti parameter C dan pilihan kernel, untuk mendapatkan hasil yang optimal.
3. Tidak intuitif untuk diinterpretasikan secara langsung seperti KNN.

2.1.2.4 Matriks Confusion

Matriks Confusion adalah sebuah alat evaluasi yang digunakan dalam analisis klasifikasi pada *machine learning*. Matriks ini menyajikan performa model dengan membandingkan hasil prediksi dengan kelas sebenarnya dari data yang diamati. Dalam matriks ini, hasil prediksi dibagi menjadi empat kategori: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative*



(FN).

Pakar di bidang ini, Dr. Adi Wibowo dari Universitas Indonesia, menggambarkan pentingnya *matriks confusion* dengan mengatakan, "*Matriks Confusion* memberikan gambaran yang jelas tentang seberapa baik model

klasifikasi dapat mengidentifikasi kelas-kelas yang ada dalam dataset. Dengan memahami hasil prediksi model dalam konteks *matriks confusion*, kita dapat mengevaluasi keakuratan dan kinerja keseluruhan dari model tersebut." Pernyataan ini menunjukkan bagaimana *matriks confusion* digunakan sebagai alat yang penting dalam evaluasi performa model klasifikasi, serta pentingnya interpretasi hasil prediksi dalam konteks *matriks confusion*.

2.1.3 CRISP-DM

CRISP-DM adalah singkatan dari *Cross Industry Standard Process for Data Mining*. Para ahli di Indonesia juga telah memberikan pandangan mereka tentang CRISP-DM. Sebagai contoh, Profesor Bambang Riyanto dari Universitas Indonesia menyatakan, "CRISP-DM memberikan pendekatan yang sistematis dan terstruktur dalam mengelola proyek data mining, sehingga membantu organisasi untuk memahami dan memanfaatkan potensi data mereka dengan lebih efektif." Pernyataan ini mencerminkan pentingnya CRISP-DM dalam konteks pengembangan proyek data mining di Indonesia, serta pengakuan atas manfaatnya dalam mendukung pengambilan keputusan yang berbasis data.

2.1.3.1 Business Understanding

Mengidentifikasi tujuan bisnis untuk membangun model prediksi obesitas dengan tujuan membantu pengelola gym dalam menilai risiko obesitas pada anggota mereka. Menentukan akurasi prediksi yang tinggi antara algoritma *KNN* dan *SVM* dalam kemampuan untuk mengklasifikasikan obesitas dengan baik, dan kemudahan interpretasi model.

2.1.3.2 Data Understanding

Mengumpulkan data yang relevan seperti data antropometri, riwayat kesehatan, dan aktivitas fisik anggota gym. Melakukan eksplorasi data untuk mengidentifikasi pola dan distribusi dalam data, serta mengetahui korelasi antara fitur-fitur yang ada. Mengevaluasi kualitas data dengan melakukan pemeriksaan keberadaan missing values, outliers, dan data yang tidak konsisten.

2.1.3.3 Data Preparation

Melakukan imputasi nilai yang hilang, penanganan outliers, dan normalisasi data jika diperlukan. Selanjutnya memilih fitur-fitur yang paling relevan untuk prediksi obesitas. Bagi data menjadi subset pelatihan dan pengujian dengan proporsi yang sesuai.

2.1.3.4 Modeling

- a. Implementasi model *KNN*: Latih model *KNN* dengan berbagai nilai k untuk menentukan yang optimal.
- b. Implementasi model *SVM*: Latih model *SVM* dengan berbagai kernel (misalnya, *linear*, *polinomial*, atau *RBF*) dan parameter C untuk menemukan yang terbaik.

2.1.3.5 Evaluation

Evaluasi model *KNN* dan *SVM* menggunakan metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score* pada subset pengujian. Bandingkan kinerja kedua model dan identifikasi model yang memberikan hasil terbaik untuk prediksi obesitas.

2.1.3.6 Deployment

Menerapkan model terbaik ke dalam lingkungan produksi *Destroyer Muscle Gym*. Berikan pelatihan kepada staf gym tentang cara menggunakan model untuk menilai risiko obesitas pada anggota. Pantau kinerja model secara berkala dan lakukan evaluasi ulang jika diperlukan.

2.1.4 Waterfall

Metode Waterfall pertama kali diperkenalkan oleh Dr. Winston W. Royce pada tahun 1970. Metode ini mengusulkan pendekatan pengembangan perangkat lunak yang sistematis dan berurutan. Setiap tahap dalam metode ini harus diselesaikan sepenuhnya sebelum tahap berikutnya dimulai, mirip dengan air terjun yang mengalir dari satu tingkat ke tingkat berikutnya. Tahapan-tahapan dalam metode Waterfall meliputi:

2.1.4.1 Requirement Analysis (Analisis Kebutuhan)

Pada tahap ini, kebutuhan pengguna dan sistem diidentifikasi dan didokumentasikan. Tujuannya adalah untuk memahami apa yang diinginkan dan dibutuhkan oleh pengguna serta bagaimana sistem harus berfungsi. Analisis ini biasanya melibatkan wawancara, survei, dan studi dokumen terkait.

2.1.4.2 System Design (Desain Sistem)

Setelah kebutuhan dianalisis dan dipahami, tahap berikutnya adalah merancang arsitektur sistem yang akan dibangun. Desain ini mencakup pemilihan teknologi, perancangan database, antarmuka pengguna, dan komponen sistem lainnya. Desain sistem yang baik sangat penting untuk memastikan implementasi yang sukses.

2.1.4.3 Implementation (Implementasi)

Pada tahap ini, desain yang telah dibuat diterjemahkan ke dalam kode program. Pengembang menulis kode dan membuat modul-modul sistem sesuai dengan spesifikasi yang telah dirancang. Implementasi harus dilakukan dengan hati-hati untuk menghindari kesalahan dan memastikan bahwa semua kebutuhan telah terpenuhi.

2.1.4.4 Testing (Pengujian)

Setelah sistem dikembangkan, tahap selanjutnya adalah pengujian untuk memastikan bahwa sistem bekerja sesuai dengan yang diharapkan. Pengujian mencakup pengujian unit, pengujian integrasi, pengujian sistem, dan pengujian penerimaan. Tujuannya adalah untuk menemukan dan memperbaiki bug serta memastikan bahwa sistem memenuhi kebutuhan pengguna.

2.1.4.5 Deployment (Penerapan)

Setelah sistem lulus pengujian, tahap berikutnya adalah penerapan sistem ke lingkungan produksi. Ini melibatkan instalasi perangkat lunak, migrasi data, dan pelatihan pengguna. Penerapan harus dilakukan dengan hati-hati untuk meminimalkan gangguan terhadap operasi bisnis.

2.1.4.6 Maintenance (Pemeliharaan)

Tahap terakhir adalah pemeliharaan sistem. Setelah sistem diterapkan, perlu dilakukan pemantauan dan pemeliharaan untuk memastikan sistem tetap berjalan dengan baik dan memenuhi kebutuhan pengguna yang mungkin berubah seiring waktu. Pemeliharaan mencakup perbaikan bug, peningkatan fitur, dan adaptasi terhadap perubahan lingkungan.

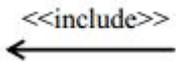
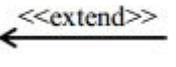
2.1.5 Pemodelan Sistem UML

2.1.5.1 Use Case Diagram

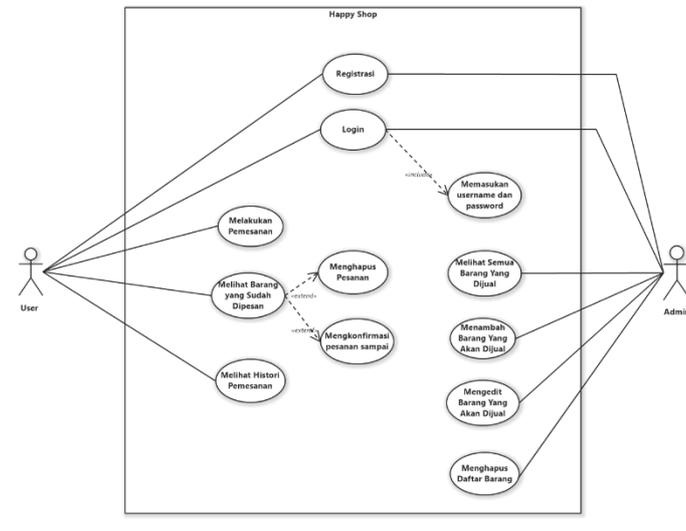
Menurut Martin Fowler (2003) mengatakan “Use Case Diagram adalah teknik merekam persyaratan fungsional sebuah sistem”. Use case mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan member sebuah narasi tentang bagaimana sistem tersebut digunakan.

Table 2. 1 Use Case Diagram

Komponen UseCase	Keterangan
 Aktor	Merupakan sebuah komponen yang menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lainnya) yang berinteraksi dengan sistem
 Use Case	Use case adalah gambaran fungsional dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.
 Association	Abstraksi hubungan antara aktor dan UseCase
 Generalisasi	Menunjukkan spesialisasi aktor untuk berpartisipasi dengan UseCase

	Menunjukkan bahwa suatu Use Case seluruhnya merupakan fungsionalitas dari usecase lainnya.
	Menunjukkan bahwa suatu usecase merupakan tambahan fungsional dari Use Case lainnya jika suatu kondisi sudah terpenuhi

Dari penjelasan diatas berikut adalah contoh gambar untuk usecase diagram



2.1.5.2 Activity Diagram

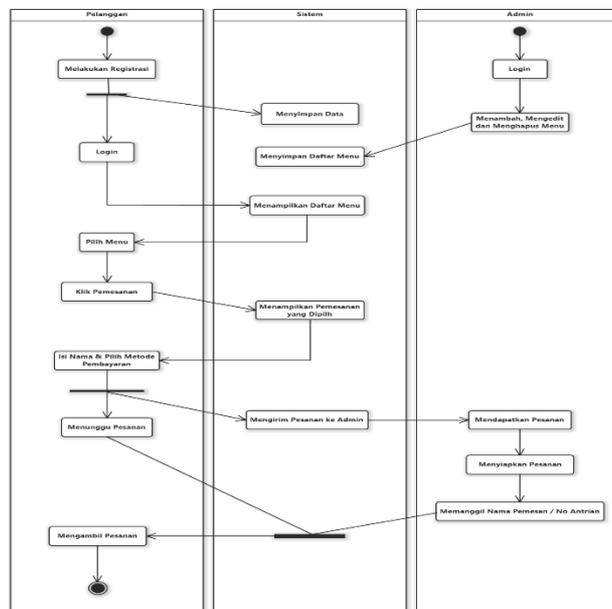
Menurut Martin Fowler (2004:163) “Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis dan jalur kerja”.

Table 2. 2 Activity Diagram

Activity Diagram	Keterangan
------------------	------------

 Status Awal	Start State, sebagai tanda awal proses dari activity diagram
 Activity	Menampung event atau aktivitas pada proses sistem.
 Decision	Decision, digunakan ketika terjadi pemilihan 2 kondisi event pada diagram
 Join	Penggabungan dimana lebih dari satu aktivitas lalu digabungkan menjadi satu
 Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
 Status Akhir	End State, sebagai tanda akhir dari activity diagram.

Dari penjelasan diatas berikut adalah contoh gambar untuk activity diagram



2.1.6 Pemrograman Sistem

2.1.6.1 Python

Python adalah bahasa pemrograman tingkat tinggi yang didesain dengan fokus pada keterbacaan dan sintaks yang jelas. Diciptakan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, Python telah menjadi salah satu bahasa pemrograman yang paling populer dan digunakan secara luas di berbagai bidang seperti pengembangan web, analisis data, kecerdasan buatan, ilmu data, dan banyak lagi.

Python dikenal dengan filosofi "*batteries included*" yang berarti bahwa bahasa ini dilengkapi dengan berbagai pustaka dan modul standar yang kaya fitur, memungkinkan pengguna untuk dengan cepat membangun berbagai macam aplikasi tanpa harus menulis kode dari awal. Python juga dikenal dengan kemudahan pembelajarannya dan komunitas yang aktif, membuatnya menjadi pilihan yang populer bagi pemula maupun profesional dalam dunia pemrograman.

2.1.6.2 Html

HTML, atau *Hypertext Markup Language*, adalah bahasa markup yang digunakan untuk membuat dan merancang halaman web. Dengan menggunakan elemen-elemen markup seperti tag dan atribut, HTML memungkinkan pengguna untuk menentukan struktur dasar dan mengatur konten dalam sebuah halaman web. Ini adalah bagian integral dari *World Wide Web* dan digunakan oleh peramban web untuk menampilkan halaman-halaman web dengan cara yang diinginkan oleh pengembang. HTML juga mendukung penggunaan hyperlink, gambar, tabel, formulir, dan berbagai elemen lainnya untuk memperkaya konten halaman web.

2.1.6.3 Flask

Flask adalah sebuah framework web mikro yang ditulis dalam bahasa pemrograman Python. Ini dirancang untuk membuat aplikasi web dengan cepat dan mudah, dengan fokus pada kesederhanaan dan fleksibilitas. Dengan Flask, pengembang dapat membuat aplikasi web yang ringan dan efisien, baik itu sebuah blog sederhana, *API RESTful*, atau bahkan aplikasi web yang lebih kompleks. Salah satu keunggulan Flask adalah kemampuannya untuk diintegrasikan dengan mudah dengan berbagai macam ekstensi dan library Python lainnya. Ini memberikan para pengembang kontrol penuh atas struktur dan perilaku aplikasi web mereka.

2.1.6.4 MySQL

W3Schools, sebagai sumber referensi dan pembelajaran online yang populer, mendeskripsikan *MySQL* sebagai sebuah sistem manajemen basis data relasional (RDBMS) yang menggunakan bahasa kueri *SQL (Structured Query Language)* untuk mengelola dan memanipulasi data dalam sebuah database. *W3Schools* mungkin menekankan bahwa *MySQL* adalah perangkat lunak sumber terbuka yang serbaguna, dengan kemampuan untuk digunakan dalam berbagai aplikasi, termasuk situs web dinamis, aplikasi bisnis, dan sistem informasi. Mereka juga mungkin menyoroti tutorial dan referensi yang tersedia di situs mereka untuk membantu pengguna mempelajari dan menguasai *MySQL*.

2.2 Kajian Penelitian Terdahulu

2.2.1 Paper 1

Perbandingan Algoritma *Support Vector Machine* dan *KNN* dalam Memprediksi Struktur Sekunder Protein

- **Penulis** : Anggi Tasari, Dewan Dinata Tarigan, Erika Nia Devina Br Purba, Kana Saputra S
- **Publikasi** : JURNAL INFORMATIKA, Volume 9, Nomor 2, Tahun 2022
- **Akreditasi** : Sinta 4
- **Tujuan Penelitian** : Membandingkan kinerja algoritma *Support Vector Machine (SVM)* dan *K-Nearest Neighbors (KNN)* dalam memprediksi struktur sekunder protein. Struktur sekunder protein, seperti heliks alfa, lembar beta, dan loop, memiliki peran penting dalam memahami fungsi dan karakteristik protein, oleh karena itu, penelitian ini bertujuan untuk mengevaluasi keefektifan kedua algoritma dalam memprediksi struktur tersebut.
- **Metodologi** : Penelitian ini menggunakan dataset yang berisi informasi struktur sekunder protein yang telah ditandai secara manual. Selanjutnya, penelitian ini menerapkan algoritma *Support Vector Machine (SVM)* dan *K-Nearest Neighbors (KNN)* untuk memprediksi struktur sekunder protein. *SVM* adalah algoritma pembelajaran mesin yang memisahkan dua kelas dengan mencari hyperplane terbaik, sedangkan *KNN* adalah algoritma klasifikasi yang memprediksi kelas suatu titik data berdasarkan mayoritas kelas dari k tetangga terdekatnya.
- **Temuan Utama** : Hasil penelitian menunjukkan bahwa algoritma *Support Vector Machine (SVM)* memiliki kinerja yang lebih baik daripada *K-Nearest Neighbors (KNN)* dalam memprediksi struktur sekunder protein. *SVM* mampu memberikan prediksi yang lebih akurat dengan tingkat keberhasilan yang lebih tinggi dibandingkan dengan *KNN*. Hal ini menunjukkan bahwa *SVM*

dapat menjadi pilihan yang lebih unggul dalam konteks memprediksi struktur sekunder protein.

- **Keterbatasan Penelitian** : Validasi model yang digunakan untuk mengevaluasi kinerja *SVM* dan *KNN* mungkin tidak cukup kuat. Metode validasi silang yang lebih kompleks atau validasi menggunakan dataset eksternal mungkin diperlukan untuk memastikan generalisasi yang lebih baik dari model.

2.2.2 Paper 2

Analisis Prediksi Level Obesitas Menggunakan Perbandingan Algoritma *Machine Learning* dan *Deep Learning*

- **Penulis** : Setiyani L, Nur Indahsari A, Roestam R
- **Publikasi** : JTERA (Jurnal Teknologi Rekayasa), Vol. 8, No. 1, Juni 2023
- **Akreditasi** : Sinta 3
- **Tujuan Penelitian** : Menganalisis dan membandingkan kinerja algoritma *Machine Learning* dan *Deep Learning* dalam memprediksi level obesitas.
- **Metodologi** : Metode yang digunakan meliputi pengumpulan data tingkat obesitas, pemrosesan data, pembagian data menjadi set pelatihan dan set pengujian, implementasi algoritma *Machine Learning* dan *Deep Learning*, serta evaluasi kinerja menggunakan metrik yang relevan seperti akurasi, presisi, dan *recall*.
- **Temuan Utama** : Hasil penelitian menunjukkan bahwa kedua pendekatan, baik *Machine Learning* maupun *Deep Learning*, mampu memberikan prediksi level obesitas yang cukup akurat. Namun, *Deep Learning* cenderung

memberikan kinerja yang sedikit lebih baik dalam hal akurasi prediksi dibandingkan dengan *Machine Learning*.

- **Keterbatasan Penelitian** : Tidak semua model machine learning atau *deep learning* mungkin cocok untuk tugas prediksi level obesitas. Kesesuaian model dengan masalah yang dihadapi perlu dievaluasi dengan cermat.

2.2.3 Paper 3

Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode *Naïve Bayes* Dan *Support Vector Machine (SVM)*

- **Penulis** : Fluorida Fibrianda M, Bhawiyuga A
- **Publikasi** : Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 2, No. 9, September 2018
- **Akreditasi** : Sinta 3
- **Tujuan Penelitian** : Menganalisis dan membandingkan akurasi deteksi serangan pada jaringan komputer menggunakan metode *Naïve Bayes* dan *Support Vector Machine (SVM)*.
- **Metodologi** : Tahapan klasifikasi serangan menggunakan metode behavior based membutuhkan sebuah dataset dan metode. Dengan melakukan perbandingan pola atau aktivitas yang ada pada sebuah data, kemudian dilakukan klasifikasi dengan sebuah metode dan menghasilkan sebuah model.
- **Temuan Utama** : Performa yang dihasilkan dari confusion matrix pada masing-masing classifier *Naïve Bayes*, *SVM Linear*, *SVM Polynomial*, dan *SVM Sigmoid* menghasilkan persentase akurasi berturut-turut sebesar 85,055%, 99,995%, 99,999% dan 99,995%.

- **Keterbatasan Penelitian** : Penilaian performa model hanya dilakukan berdasarkan akurasi dan kurva *ROC*. Penggunaan metrik evaluasi lainnya, seperti presisi, *recall*, atau *F1-score*, dapat memberikan gambaran yang lebih lengkap tentang kinerja model, terutama dalam kasus yang tidak seimbang (*imbalanced*) seperti ini.