# BAB II

# TINJAUAN PUSTAKA

## 2.1 Penelitian Terkait

Berikut ini merupakan landasan teori yang meninjau atau mengkaji dari sumber acuan yang relavan dengan mengutamakan kepada jurnal ilmiah yang telah di publikasikan terlebih dahulu baik secara nasional maupun internasional yang berfungsi sebagai memperdalam dan menimbulkan permasalahan yang telah di kaji atau di teliti.

Sebagai bahan rujukan dalam membangun pengembangan model klasifikasi program kerja terbaik menggunakan algoritma SVM dengan fungsi kernel *sigmoid* dan *perceptron* menggunakan fungsi aktivasi *sigmoid* ini, digunakan 14 artikel penelitian yang terbagi atas 7 artikel klasifikasi SVM, dan 7 artikel klasifikasi *perceptron*.

Berikut ini pada Tabel II.1 menampilkan rujukan rangkuman artikel penelitian yang mengangkat masalah klasifikasi SVM.

Tabel II.1 Penelitian Terkait Klasifikasi SVM

| No | Penulis | Judul | Tahun | Hasil Penelitian |
|---|---|---|---|---|
| 1 | Min-Wei Huang, Chih-Wen Chen, Wei-Chao Lin, Shih-Wen Ke, Chih-Fong Tsai | SVM and SVM Ensembles in Breast Cancer Prediction | 2017 | pengklasifikasi SVM pemilihan akurasi klasifikasi 98,28% |

Tabel II.1 Penelitian Terkait Klasifikasi SVM (Lanjutan)

| No | Penulis | Judul | Tahun | Hasil Penelitian |
|---|---|---|---|---|
| 2 | N. Krishnaveni, Dr. V. Radha | Ensemble of Multiple Kernel SVM Classifiers for Detection of Online Spam Reviews | 2020 | SVM klasifikasi. Evaluasi kinerja dilakukan berdasarkan pada parameter seperti Accuracy, Precision, Recall, F-Measure dan nilai akurasi 88.20% |
| 3 | Ashok Bekkanti, Shilpa Itnal, VSRK Prasad Gunde, Gayatri Parasa, CMAK Zeelan Basha | Computer Based Classification of Diseased Fruit using K-Means and Support Vector Machine | 2020 | Klasifikasi SVM dengan mendapatkan hasil dari akurasi 92.23% |
| 4 | Woro Isti Rahayu, Aulyardha Anindita, Mohamad Nurkamal Fauzan | Penentuan Validasi Data Pemilih Dan Klasifikasi Hasil Pemilu DPRD Kab. Bone Untuk Memprediksi Partai Pemenang Menggunakan Metode Naive Bayes | 2020 | SVM dan Naïve bayes melakukan klasifikasi dimana SVM mendapatkan akurasi 93.00% |

Tabel II.1 Penelitian Terkait Klasifikasi SVM (Lanjutan)

| No | Penulis | Judul | Tahun | Hasil Penelitian |
|---|---|---|---|---|
| 5 | Gunawan, Yuza Reswan | Desain Aplikasi Pengenalan Pola Tanda Tangan Menggunakan Metode *Support Vector Machine* (SVM) | 2021 | Klasifikasi SVM dengan mendapatkan hasil dari akurasi 89.20% |
| 6 | Md. Imran Hossain Showrov, Muhammad Tanveer Islam, Md. Dulal Hossain, Md. Shakil Ahmed | Performance Comparison Of Three Classifiers For The Classification Of Breast Cancer Dataset | 2019 | Klasifikasi SVM model *sigmoid* mendapatkan hasil akurasi 30.88% |
| 7 | Intisar Shadeed Al-Mejibli, Dhafar Hamed Abd, Jwan K.Alwan, Abubaker Jumaah Rabash | Performance Evaluation of Kernels in Support Vector Machine | 2018 | Klasifikasi SVM model *sigmoid* mendapatkan hasil akurasi 99.3% |
| 8 | Utomo Pujianto, Ilham Ari Elbaith Zaeni, Ninon Oktaviani Irawan | SVM Method for Classification of Primary School Teacher Education Journal Articles | 2019 | Mendapatkan nilai akurasi dari hasil klasifikasi SVM sebesar 80.56% |

Tabel II.1 Penelitian Terkait Klasifikasi SVM (Lanjutan)

| No | Penulis | Judul | Tahun | Hasil Penelitian |
|---|---|---|---|---|
| 9 | Trifebi Shina Sabrila, Yufis Azhar, Christian Sri Kusuma Aditya | Analisis Sentimen *Tweet* Tentang UU Cipta Kerja Menggunakan Algoritma SVM Berbasis PSO | 2022 | Hasil yang di dapatkan dari klasifikasi SVM sebesar 92.99% |

Berikut ini pada Tabel II.2 menampilkan rujukan rangkuman artikel penelitian yang mengangkat masalah klasifikasi *perceptron.*

Tabel II.2 Penelitian Terkait Klasifikasi *Perceptron*

| No | Penulis | Judul | Tahun | Hasil Penelitian |
|---|---|---|---|---|
| 1 | I M D Maysanjaya | Comparative study of classification method on diagnosis of plasmodium phase | 2020 | Hasil klasifikasi *perceptron* dengan nilai akurasi sebesar 81.08% |
| 2 | Shadi Diab | Optimizing Stochastic Gradient Descent in Text Classification Based on Fine-Tuning Hyper-Parameters Approach | 2018 | Hasil klasifikasi *perceptron* dengan nilai akurasi sebesar 80.56% |

Tabel II.2 Penelitian Terkait Klasifikasi *Perceptron* (Lanjutan)

| No | Penulis | Judul | Tahun | Hasil Penelitian |
|---|---|---|---|---|
| 3 | Riyanto Sigit, Elvi Triyana, Mochammad Rochmad | Cataract Detection Using Single Layer Perceptron Based on Smartphone | 2019 | Hasil klasifikasi *perceptron* dengan nilai akurasi sebesar 85,30% |
| 4 | Andiko Prasetyo, Dessy Ana Laila Sari | Implementasi Jaringan Syaraf Tiruan Model Perceptron Untuk Klasifikasi Karir Alumni SMKN 1 Glagah Tahun Angkatan 2018 | 2022 | pengklasifikasi *perceptron* dengan nilai akurasi klasifikasi 87,38% |
| 5 | Syarifuddin N. Kapita, Samlan Mahdi, Firman Tempola | Penilaian Pengetahuan Siswa Dengan Jaringan Syaraf Tiruan Algoritma Perceptron | 2020 | Mendapatkan nilai akurasi tertinggi sebesar 96.20% |
| 6 | Yohanes Pangaribuan, Masdiana Sagala | Menerapkan Jaringan Saraf Tiruan untuk Mengenali Pola Huruf Menggunakan Metode Perceptron | 2017 | Mendapatkan nilai akurasi tertinggi sebesar 92.3% |

Tabel II.2 Penelitian Terkait Klasifikasi *Perceptron* (Lanjutan)

| No | Penulis | Judul | Tahun | Hasil Penelitian |
|---|---|---|---|---|
| 7 | Syafei Karim | Perubahan perilaku Non-Player Character (NPC) pada Game Arabic Hunter menggunakan Jaringan Syaraf Tiruan Perceptron | 2018 | Mendapatkan nilai akurasi tertinggi sebesar 93.2% |
| 8 | Riyanto Sigit, Elvi Triyana, Mochammad Rochmad | Cataract Detection Using Single Layer Perceptron Based on Smartphone | 2019 | Mendapatkan nilai akurasi tertinggi sebesar 85% |
| 9 | Dimas Agung Riansa, Widodo, Bambang Prasetya Adhi | Pengenalan Tanda Tangan Menggunakan Algoritma Single Layer Perceptron | 2019 | Mendapatkan nilai akurasi tertinggi sebesar 78.66% |

Berdasarkan rangkuman artikel penelitian terkait yang klasifikasi SVM pada Tabel II.1 dan *Perceptron* pada Tabel II.2, dibentuk tabulasi hasil evaluasi dari penelitian tersebut seperti terlihat pada Tabel II.3

Tabel II.3 Evaluasi Perbandingan Klasifikasi SVM Dan *Perceptron*

| **Penelitian** | **Nilai Akurasi** | **Nilai Akurasi** | **Penelitain** |
|---|---|---|---|
| Penelitian SVM 1 | 98,28% | 81,08% | Penelitian *Perceptron* 1 |
| Penelitian SVM 2 | 88,20% | 80,56% | Penelitian *Perceptron* 2 |
| Penelitian SVM 3 | 92,23% | 85,30% | Penelitian *Perceptron* 3 |
| Penelitian SVM 4 | 93,00% | 87,38% | Penelitian *Perceptron* 4 |
| Penelitian SVM 5 | 89,20% | 96,20% | Penelitian *Perceptron* 5 |
| Penelitian SVM 6 | 30,88% | 92,3% | Penelitian *Perceptron* 6 |
| Penelitian SVM 7 | 99,3% | 93,2% | Penelitian *Perceptron* 7 |
| Penelitian SVM 8 | 80,56% | 85,0% | Penelitian *Perceptron* 8 |
| Penelitian SVM 9 | 92,99% | 78,6% | Penelitian *Perceptron* 9 |
| Nilai Akurasi Tertinggi | 99,3% | 96,20% | Nilai Akurasi Tertinggi |

Didapatkan dari hasil pada pembahasan Tabel II.1 pada klasifikasi SVM dan Tabel II.2 klasifikasi *perceptron* bahwasanya untuk model algoritma SVM lebih memiliki nilai *Accuracy* lebih tinggi untuk masalah klasifikasi dalam perbandingan model dengan dari hasil jurnal terkait. Maka dari itu peneliti ingin melakukan pengujian dengan kasus terbaru melakukan klasifikasi antara model SVM dan

*perceptron* untuk mengetahui dalam kasus yang di angkat model mana yang memiliki nilai tertinggi dan layak di uji cobakan.

## 2.2 Program Kerja

Program kerja merupakan kegiatan untuk menentukan sebuah tujuan berorganisasi dengan membuat berbagai rencana sebagai penjabaran visi, misi, kebijakan dan program pembangunan yang akan dilaksanakan sebagai pedoman kerja bagi pegawai dalam memenuhi tuntutan visi dan misi perusahaan ke depannya. Perencanaan program kerja merupakan pondasi dari manajemen, karena merupakan proses untuk memutuskan tujuan apa yang akan dikejar selama suatu jangka waktu yang akan datang dan apa yang harus dilakukan agar tujuan itu dapat tercapai (Suradi, 2015).

Jenis dari program kerja dapat ditinjau dari berbagai aspek, di antaranya adalah (Sholichah, 2018) :

1. Tujuan

   Terdiri dari program kerja dengan tujuan mencari keuntungan dan program kerje sukarela.

2. Jenis

   Terdiri dari program kerja pendidikan, program kerja kemasyarakatan, dan program kerja lain yang bergantung dari isi program kerja tersebut.

3. Jangka waktu

   Terdiri dari program kerja jangka waktu pendek, jangka menengah, dan jangka panjang.

4. Keluasan

   Terdiri dari program kerja sempit (yang cakupannya terbatas) dan program kerja luas (yang cakupannya memiliki banyak variabel).

5. Pelaksanaannya

   Terdiri dari program kerjakecil (dilaksanakan oleh beberapa orang saja) dan program kerja besar (dilaksanakan oleh orang banyak).

6. Sifatnya

   Terdiri dari program kerja penting (yang dampaknya menyangkut hal-hal yang vital) dan program kerja kurang penting (yang menyangkut hal-hal yang kurang vital).

Untuk mengetahui kinerja suatu program kerja, perlu dilakukan evaluasi dengan cara membandingkannya dengan kriteria yang telah ditentukan atau tujuan yang ingin dicapai dengan hasil yang dicapai. Hasil dalam bentuk informasi ini kemudian digunakan sebagai bahan pertimbangan untuk pembuatan keputusan dan penentuan kebijakan selanjutnya.

Evaluasi terhadap sebuah program kerja harus dilakukan secara sistematis dengan melalui proses pengumpulan dan analisis data yang dapat dipertanggungjawabkan kebenarannya, untuk mengetahui tingkat keberhasilan suatu program kerja (Sofyan, Setiyadi, Harlina Harja, & Sari, 2020).

## 2.3 Machine Learning

*Machine learning* merupakan bagian dari *Aritificial Intelligence* (AI) yang ide dasarnya adalah bagaimana mesin dapat belajar dari data, mengidentifikasi pola-pola yang ditemukan pada data tersebut, dan kemudian menarik sebuah

keputusan dengan campur tangan manusia seminim mungkin. Metode pada *machine learning* dapat dikategorikan menjadi dua jenis, yaitu (Chahal & Gulia, 2019) :

1. *Shallow Learning*

   Metode yang pada umumnya menggunakan *neural network* dengan jumlah *layer* satu atau dua. Metode ini terbagi menjadi beberapa bagian, seperti *supervised learning, unsupervised learning, semi-supervised learning,* dan *reinforcement learning*.

2. *Deep Learning*

   Metode yang pada umumnya menggunakan *neural network* dengan jumlah *layer* lebih dari dua. Metode ini terbagi menjadi beberapa jenis model, seperti *generative models, discriminative models,* dan *hybrid models*.

*Machine learning* merupakan bagian dari metode *data mining*, dimana *data mining* itu sendiri merupakan sebuah metode untuk memproses data yang diterima dalam jumlah yang besar, dan mengekstrak informasi dari data tersebut dengan menggunakan proses cerdas dan otomatis melalui pemanfaatan algoritma dan perangkat lunak komputer (Taha Chicho, Mohsin Abdulazeez, Qader Zeebaree, & Assad Zebari, 2021).

*Machine learning* memiliki peranan yang vital dalam masalah klasifikasi. Klasifikasi itu sendiri merupakan sebuah kategori di dalam *machine learning* yang termasuk ke dalam konsep *supervised learning*. Secara umum, klasifikasi di dalam *machine learning* bertujuan untuk mengkategorikan sekelompok data ke dalam

kelas-kelas yang berbeda berdasarkan label target yang sudah ditentukan sebelumnya (Iqbal & Yadav, 2020).

Dalam prosesnya, secara umum *machine learning* melalui tahapan ekstraksi fitur data, pemilihan algoritma yang sesuai dengan masalah, proses *training* dan *testing* terhadap data yang diinputkan, dan menggunakan model hasil pembelajaran untuk menghasilkan solusi masalah dalam bentuk informasi-informasi penting dari dalam data yang diberikan (Chahal & Gulia, 2019).

Dalam *machine learning*, terdapat beberapa algoritma yang sering digunakan di dalam penelitian, antara lain (Çelik, 2018) :

1. *Artificial Neural Network*
2. *Decision Tree*
3. *Support Vector Machines*
4. *Naïve Bayes*
5. *Logistic Regression*
6. *K-Nearest Neighbors*

Kelebihan dari *machine learning* adalah dapat memberikan hasil yang lebih akurat jika data yang digunakan berjumlah banyak. Dengan kata lain, semakin banyak data yang digunakan, semakin tinggi hasil akurasi dari *machine learning* dan jika sebaliknya akan menghasilkan hasil akurasi yang semakin rendah (Hartono, Sitompul, Tulus, & Nababan, 2018). Selain itu, sifat *machine learning* yang konstruktif di dalam prosesnya menghasilkan sebuah hasil prediksi dan analisis yang akurat (Kohsasih & Situmorang, 2022).

## 2.4 Data Mining

*Data mining* merupakan sebuah step dan algoritma untuk melaksanakan kegiatan yang di peruntukkan untuk melakukan pencarian informasi yang berguna yang dapat dimanfaatkan untuk mendapatkan kepada pola yang sebelumnya yang masih tidak di ketahui bagaimana dari jumlah besar pada pekumpulan data sebagai dari inti sari untuk *knowledge discovery in database* (KDD), dengan begitu *data mining* juga dapat melakukan pemprosesan untuk melakukan pengelompokan oraganisasian dan pengindetifikasian kepada data secara otomatis dengan menemukan pola pada sekumpulan data dengan kuantitas besar dan berkompleks (Alkhairi & Situmorang, 2022).

Dalam data mining sendiri untuk melakukan klasifikasi yang sudah termasuk ke dalam kategori dari *supervised* learning dan juga dapat digunakan kepada melakukan mengelompokkan sekumpulan data dengan terlebih dahulu pastinya memberikan penanda dengan memberikan *sub subject* label terhadap kelas target yang diinginkan setelah itu kemudian menggunakannya sebagai acuan untuk memprediksi pola-pola data berikutnya (Prasetya & Ridwan, 2019).

*Data mining* juga dapat digunakan untuk mengklasifikasikan data dalam kelompok (*cluster*) yang memiliki kedekatan tingkat kesamaan menggunakan metode *clustering* yang merupakan bagian dari *unsupervised learning* tanpa adanya pemberian label kepada kelas target yang diinginkan (Suliman, 2021).

Terlepas dari itu pastinya di awal sebelumnya melakukan mengekstrak pengetahuan yang bermanfaat dari sebuah sekumpulan data dan juga perlu dipahami secara keseluruhan kepada pendekatan apa yang akan untuk digunakan.

Mengetahui algoritma yang digunakan untuk analisis data tidaklah cukup untuk menjamin keberhasilan proses penambangan data.

Dalam membangun model yang baik, ada beberapa tahapan yang wajib untuk diterapkan dalam proses *data mining*, seperti (Said, Hassan, & Fawzy, 2018):

1. Data *preparation*

   Merupakan teknik pra-pemrosesan data yang diterapkan sebelum proses penambangan data proses ini secara keseluruhan mempengaruhi pola-pola data yang dihasilkan dan waktu yang dibutuhkan pada proses penambangan data.

2. *Data analysis* (*modeling*)

   Tahapan yang penting dalam proses penambangan data, karena disinilah dijelaskan metode-metode apa saja yang digunakan dalam menambang data tersebut. Dalam tahapan ini, biasanya beberapa metode digunakan secara simultan untuk menambang data dan bagaimana kalibrasi parameter-parameter yang digunakan dalam membangun sebuah model.

3. *Evaluation*

   Tahapan akhir dari penambangan data yang merupakan evaluasi dari model yang dihasilkan. Nilai seperti *accuracy*, *speed*, *robustness*, *scalability*, dan *interpretability* biasanya digunakan untuk mengevaluasi model penambangan data yang dihasilkan.

Teknik *data mining* secara umum terbagi menjadi tiga bagian, yaitu (Rumahorbo & Sekarwati, 2020):

1. *Clustering*

Merupakan teknik data mining yang digunakan untuk mengelompokkan data berdasarkan tingkat kemiripannya satu dengan yang lain. Berdasarkan tingkat kemiripan ini, data di kelompokkan ke dalam kelas-kelas yang disebut dengan cluster.

2. Classification

   Merupakan teknik yang paling umum digunakan di dalam data mining, dimana sekumpulan besar data diproses ke dalam sebuah model yang kemudian membaginya ke dalam kelas-kelas yang sudah ditentukan sebelumnya.

3. Regression

   Merupakan teknik data mining yang umumnya digunakan untuk masalah prediksi, dengan cara membangun model hubungan antara satu atau lebih atribut di dalam data sehingga jika ada perubahan atribut di dalam data tersebut dapat digunakan untuk memprediksikan nilainya di waktu yang akan datang.

## 2.5 Klasifikasi

Klasifikasi merupakan sebuah proses untuk menemukan sebuah pemodelan atau fungsi yang bertujuan untuk mengetahui dari memperkirakan termasuk ke dalam mana kelas data yang sebelumnya dari kelas yang tidak diketahui dengan maksud untuk menjelaskan atau membedakan konsep yang terdapat pada kelas data masing-masing (Budiman, Muliadi, & Ramadina, 2015). Metode klasifikasi sendiri menggunakan sistem pembelajaran dengan menggunakan fungsi yang berbeda

dalam memetakan setiap data yang terpilih kedalam kelompok kelas yang sudah ditetapkan sebelumnya (Hanun & Zailani, 2020).

Dalam proses klasifikasi sendiri pada data mentah yang digunakan di dalam dataset perlu dinormalisasi terlebih dahulu supaya untuk menghilangkan unsur ketidakpastian yang disebabkan oleh adanya informasi yang kurang tepat, ambiguitas dalam data masukan, tumpang tindih batas-batas antara kelas, dan ketidaktentuan dalam mendefinisikan fitur (Navastara, Julia, & Purwitasari, 2018).

Klasifikasi dapat diartikan sebagai sebuah proses untuk menentukan lokasi penempatan objek data ke dalam kategori tertentu, dengan cara mempelajari sifat-sifat, karakteristik, dan fungsi dari data tersebut, untuk kemudian dipetakan pada kelompok kelas yang telah ditetapkan sebelumnya (Hasanah, Soim, & Handayani, 2021). Pada prosesnya, klasifikasi biasanya memiliki beberapa komponen yaitu sebagai berikut (Permana & Sahara, 2019) :

1. *Class* (Kelas)

   Merupakan sebuah variabel yang terdapat di dalam model klasifikasi dan berfungsi untuk mewakili label-label yang sudah ditentukan pada objek yang akan diklasifikasikan.

2. *Predictors* (Prediktor)

   Merupakan sebuah variabel yang terdapat di dalam model klasifikasi dan berfungsi untuk mewakili karakteristik atau atribut-atribut dari data yang akan diklasifikasikan.

3. *Training Dataset* (Dataset Latih)

Merupakan kumpulan data yang digunakan untuk melatih model, sehingga model tersebut dapat menemukan pola-pola data di dalamnya.

4. *Testing Dataset* (Dataset Uji)

   Merupakan kumpulan data yang digunakan untuk menguji model, sehingga model tersebut dapat diukur kinerjanya sebelum diimplementasikan menjadi bentuk sebuah aplikasi.

Pengumpulan data merupakan hal penting yang harus diperhatikan di dalam klasifikasi, karena kualitas dari data yang diinputkan akan mempengaruhi kualitas dari hasil klasifikasi tersebut. Selain itu, pengetahuan tentang model yang akan dibangun serta pemilihan algoritma yang tepat menjadi prasyarat dalam menghasilkan keakuratan model klasifikasi yang dihasilkan (Gunawan, Zarlis, & Roslina, 2021).

Algoritma yang digunakan di dalam proses klasifikasi pada umumnya disebut sebagai *classifier*. Berbagai macam *classifier* yang dapat digunakan di dalam klasifikasi tidak dapat dibedakan begitu saja mana yang lebih superior dibandingkan dengan yang lainnya, karena setiap *classifier* menggunakan tahapan yang berbeda di dalam proses klasifikasinya. Namun, *classifier* dapat diukur performanya berdasarkan pada masalah apa *classifier* tersebut di terapkan dan bagaimana bentuk dataset yang digunakan (Shikalgar, Sawarkar, & Narwane, 2019).

Di dalam klasifikasi, masalah *overfitting* merupakan masalah klasik yang sering ditimbulkan oleh jumlah fitur pada dataset yang digunakan lebih besar daripada jumlah data latih yang digunakan, sehingga menghasilkan bias yang

menyebabkan penurunan kualitas dari data latih yang dihasilkan. Hal inilah yang menyebabkan beberapa algoritma klasifikasi menyediakan parameter-parameter yang dapat dikalibrasi secara manual ataupun otomatis, khususnya pada kasus regresi linier (Ghosh, Dasgupta, & Swetapadma, 2019).

## 2.6 Algoritma Support Vector Machine

*Support vector machine* (SVM), algoritma yang pertama kali diperkenalkan oleh Boser, Guyon, dan Vanpik pada "*The Annual Workshop on Computational Learning Theory*", merupakan algoritma yang dapat digunakan untuk menyelesaikan masalah prediksi, baik itu berupa masalah klasifikasi maupun regresi. Tujuan dari SVM adalah untuk meningkatkan kecepatan dalam proses *training* dan *testing,* sehingga dapat digunakan untuk mengolah data dalam jumlah yang besar (Tamaela, Andry Lesnussa, Y.I. Ilwaru, & Balami, 2020).

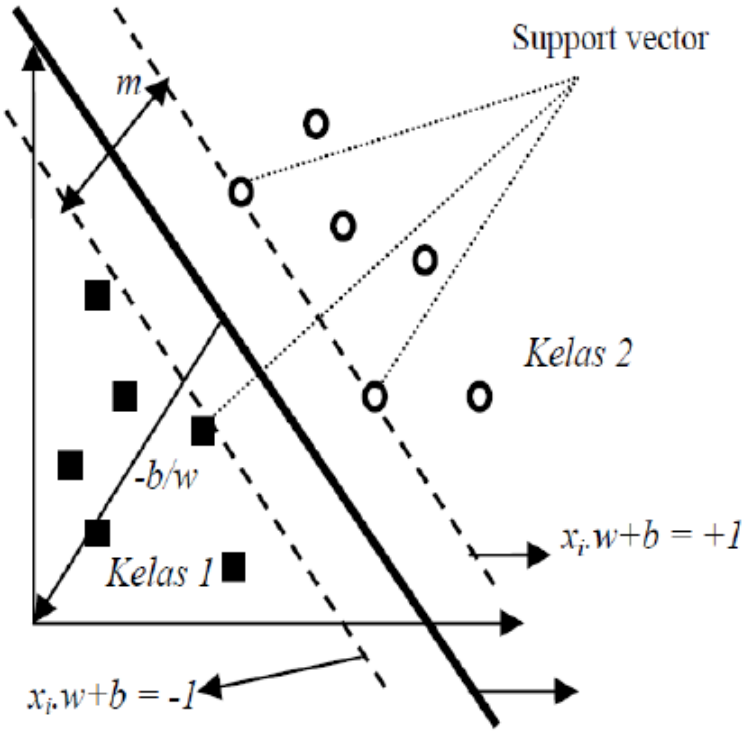
Algoritma ini sangat akurat, karena mampu untuk menangani model-model nonlinear yang kompleks. Namun, SVM kurang handal dalam hal masalah *overfitting* jika dibandingkan dengan algoritma lain yang sejenis (Nurachim, 2019).

SVM merupakan salah satu algoritma model klasifikasi yang memiliki kemampuan yang baik untuk generalisasi data yang tidak terlihat. Jika data dapat dipisahkan secara linier (*linearly seperable*), dapat ditarik keputusan dengan jumlah yang tidak terbaras dalam memisahkan data. SVM termasuk ke dalam kategori algoritma pengklasifikasi biner yang bekerja pada konsep identifikasi bidang *hyperplane*, yang memisahkan dua kelas dengan nilai margin maksimum.

SVM juga dapat menyelesaikan masalah non-linear (data yang tidak dapat dipisahkan secara linier), dengan memanfaatkan sebuah fungsi yang dinamakan

*kernel trick*. SVM yang dilengkapi dengan *kernel trick* sangat cocok digunakan untuk masalah klasifikasi karena memiliki kemampuan generalisasi yang baik (Kancherla, Bodapati, & Veeranjaneyulu, 2019).

Di dalam SVM, margin merupakan jarak antara *hyperplane* dengan titik terdekat dari masing-masing kelas. Titik-titik terdekat dari masing-masing kelas inilah yang disebut sebagai *support vector*. SVM melakukan klasifikasi data secara linier (*linearly separable*) ataupun non-linier (*nonlinear separable*), dengan menggunakan *hyperline*, seperti diilustrasikan pada Gambar II.1 (Rizal, Girsang, & Prasetiyo, 2019).



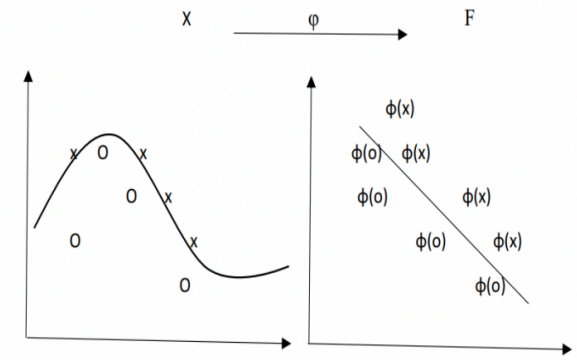

Gambar II.1 Ilustrasi Hyperplane SVM

Sumber : (Rizal et al., 2019)

Pada pemodelan pendekatan atau biasa disebut kepada dengan kernel yang dapat membantu dalam melakukan kemengatasi masalah suatu ruang fitur atau

disebutkan dengan *feature space*, dan juga berpengaruh terhadap akurasi yang akan dihasilkan. Dalam *support vector machine* (SVM) selain kernel RBF yaitu kernel linier, polinomial, dan kernel sigmoid. Dalam waktu melakukan pelatihan kernel linier lebih cepat dibandingkan dengan kernel lain dan cocok untuk data berdimensi besar. Pada kernel polinomial terdapat penggunaan derajat yang dapat diatur untuk meningkatkan kemungkinan data dapat dipisahkan secara linier dalam ruang berdimensi tinggi, tanpa memperlambat waktu model. Pada kernel sigmoid penggunaan gamma dapat diatur untuk meningkatkan nilai akurasi akan tetapi bergantung pada jumlah fitur yang digunakan (Akbar & Novrido, 2021)

Secara umum pada kasus di dunia nyata adalah kasus yang tidak linier. Untuk mengatasi sifat tidak linier dengan memakai metode kernel. Dengan metode kernel suatu data x di input space dimapping ke feature space F dengan dimensi yang lebih tinggi melalui map φ sebagai berikut φ:x→φ(x). Karena itu data x di input space menjadi φ(x) di feature space. Gambar 2 menggambarkan suatu contoh feature mapping dari ruang dua dimensi ke feature space dua dimensi. Dalam input space, data tidak bisa dipisahkan secara linier, tetapi bisa memisahkan di feature space (Alven & Endah, 2018)




Gambar II.2 Ilustrasi Hyperplane SVM Kernel

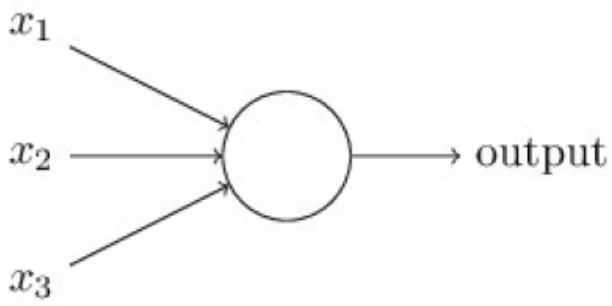
Sumber : (Diatas Penerapan Metode SVM)

Adapun rumus untuk fungsi kernel *sigmoid* dari SVM sebagai berikut :

Tanh ($\beta\ Txi + \beta i$ ) , $\beta$, $\beta i \in \Re$.

## 2.7 Algoritma Perceptron

*Perceptron* merupakan algoritma yang pertama kali dikembangkan oleh ilmuan Frank Rosenblatt pada tahun 1950an dan 1960an. *Perceptron* menerima beberapa input berbentuk binari ($x_1$, $x_2$, …, $x_n$) dan memprosesnya sehingga menghasilkan output yang juga berbentuk binari, seperti digambarkan pada Gambar II.3 (Ahamed & Akthar, 2016).




Gambar II.3 Bentuk Umum *Perceptron*

Sumber : (Ahamed & Akthar, 2016)

*Perceptron* merupakan salah satu algoritma ANN yang sederhana, dimana di dalamnya terdapat tiga lapisan, yaitu *sensory unit* (*input layer*), *associator unit* (*hidden layer*), dan *response unit* (*output layer*). Kecepatan proses algoritma *perceptron* ditentukan pula oleh laju pemahaman (*learning rate*) yang ditentukan, dimana semakin besar nilai *learning rate*, semakin sedikit cepat proses kerjanya.

Akan tetapi jika *learning rate* terlalu besar, maka akan merusak pola yang sudah benar sehingga proses pembelajaran menjadi lambat.

*Single layer Perceptron,* atau sering juga disebut dengan *perceptron,* melakukan klasifikasi dengan cara mengolah data yang diterima dari *input layer*, menghitung bobot dan bias masing-masing data pada *hidden layer*, dan mengulangi proses ini hingga hasil dari *output layer* sudah sama atau mendekati dari target (Permadi & Nugroho, 2019).

Secara umum, algoritma pembelajaran dalam *perceptron* adalah sebagai berikut (Kurniawan, Hakimah, & Agustini, 2020):

1. Inisialisasi bobot $w_j$ dan bias θ dengan bilangan yang sangat kecil.
2. Keluaran $\hat{y}$ diperoleh berdasarkan persamaan:

$$\hat{y} = f\left(\sum_{j=1}^{N} w_j x_j - \theta\right)$$

3. *Update* bobot berdasarkan aturan:

$$w_k(t+1) = w_k t + \alpha(y - \hat{y})x$$

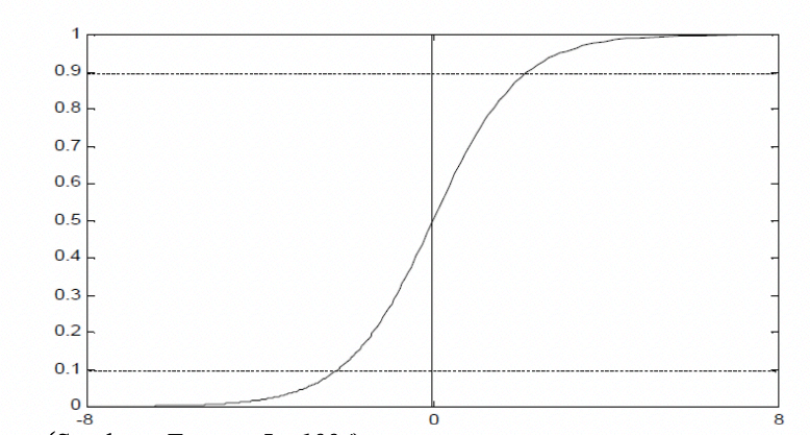
Dengan $\alpha$ adalah *learning rate*

4. Jika $\hat{y}$ = y , *perceptron* tidak berubah. Jika $\hat{y} \neq$ y, ulangi langkah 3 untuk melanjutkan *training* dan *learning*

Pada umumnya untuk melakukan dalam penerapan fungsi aktivasi merupakan fungsi yang digunakan pada jaringan saraf untuk mengaktifkan atau tidak mengaktifkan neuron. Karakteristik yang harus dimiliki oleh fungsi aktivasi jaringan perambatan balik antara lain harus kontinyu, terdiferensialkan, dan tidak menurun secara monotonis (monotonically non-decreasing). Lebih lanjut, untuk

efisiensi komputasi, turunan fungsi tersebut mudah didapatkan dan nilai turunannya dapat dinyatakan dengan fungsi aktivasi itu sendiri. Fungsi aktivasi yang di analisis adalah sigmoid biner dan sigmoid bipolar (Julpan, Erna et Al, 2016).

Adapun rumus untuk fungsi aktivasi *sigmoid biner* memiliki nilai pada range 0 sampai 1, dimana berikut gambar yang didefenisikan sebagai berikut :




Gambar II.4 Fungsi Aktivasi *sigmoid* Pada *Perceptron*

Sumber : (Julpan, Erna et Al, 2016).

Adapun rumus untuk fungsi aktivasi *sigmoid* dari *perceptron* sebagai berikut :

$$y = f(x) = \frac{1}{(1+e^{-x})}$$

## 2.8 Fungsi Aktivasi

Fungsi aktivasi adalah fungsi yang digunakan untuk menjawab masalah non-linieritas data di dalam *neural network*. Pemilihan fungsi aktivasi yang tepat sangatlah penting dalam *neural network,* karena akan mempengaruhi kinerja model yang dihasilkan (Misra, 2020). Fungsi aktivasi memanipulasi data melalui

penurunan gradien dalam menghasilkan output pada *neural network* (Bhojani & Bhatt, 2021).

Fungsi aktivasi merupakan output yang diperoleh pada sebuah *neuron* dengan menggunakan batasan aktivasi tertentu. Fungsi aktivasi berfungsi sebagai sebagai penentu nilai output yang dihasilkan dari suatu *neuron*. Beberap fungsi aktivasi yang umum digunakan di dalam *neural network* adalah fungsi ReLu, Tanh, dan Sigmoid (Euis Saraswati, Yuyun Umaidah, & Apriade Voutama, 2021).

Fungsi ReLu (*rectified liner unit*) merupakan fungsi non-linier dimana pengaktifan *neuron* tidak dilakukan secara bersamaan, dan hanya ketika output dari transformasi linier bernilai nol (Sharma, Sharma, & Anidhya, 2020). ReLU memiliki kelebihan dalam hal kemampuannya membantu mengurangi biaya komputasi, mempercepat proses pelatihan, dan mencegah gradien yang hilang (Nguyen, Pham, Ngo, Ngo, & Pham, 2021). Fungsi ini membandingkan antara nilai ambang bawah = 0. ReLu akan mengembalikan nilai 0 jika 0 < x dan mengembalikan nilai x jika sebaliknya, berdasarkan persamaan berikut (Kohsasih et al., 2022):

$$f(x) = max(0, x) \quad \text{II.12}$$

Fungsi Tahn merupakan fungsi simetris yang berbentuk s (*sigmoid*), dimana outputnya terletak pada rentang nilai -1 sampai +1. Dalam penggunaannya, fungsi Tanh dituliskan menggunakan persamaan (Chen & Wang, 2020):

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad \text{II.13}$$

Fungsi Sigmoid merupakan fungsi aktivasi yang paling umum digunakan di dalam *neural network*, karena memiliki hasil akurasi yang cukup tinggi dan mudah

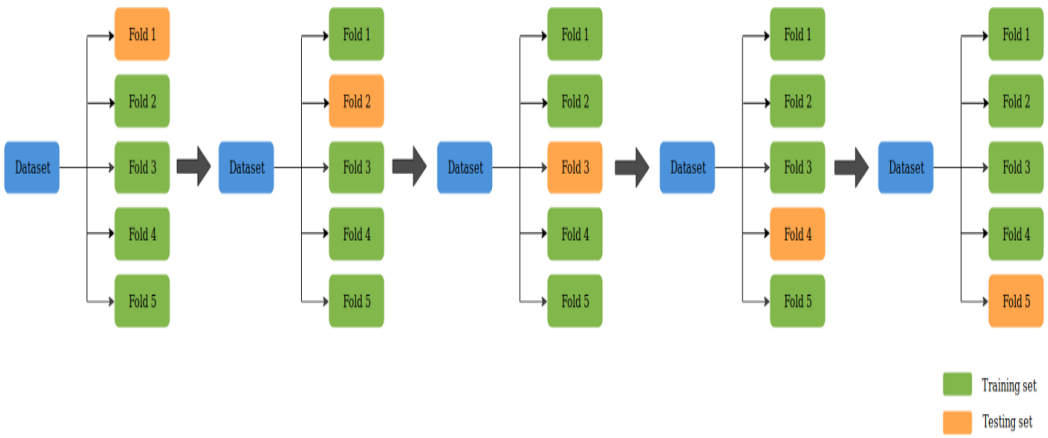
untuk dibedakan degan fungsi aktivasi yang lain. Dalam penggunaannya, fungsi Sigmoid dituliskan menggunakan persamaan (Si, Harris, & Yfantis, 2020) :

$$f(x) = \frac{1}{1+e^{-x}} \quad \text{II.14}$$

## 2.9 Cross Validation

*Cross validation,* atau K-*fold cross validation*, merupakan metode yang biasanya digunakan untuk mengevaluasi model *machine learning*. Berbeda dengan cara manual dimana data dibagi menjadi beberapa bagian untuk *data training* dan beberapa bagian untuk data *testing* dan kemudian dievaluasi dengan melihat tingkat kesalahannya, *cross validation* bekerja dengan cara membagi data menjadi *fold* berjumlah K dan memastikan bahwa setiap *fold* digunakan dalam tahapan evaluasi.

Sebagai contoh, jika digunakan nilai K = 5, yang berarti evaluasinya menggunakan teknik 5-*fold cross validation*, maka data dibagi menjadi 5 bagian serata mungkin dan kemudian dievaluasi secara bertahap dengan melakukan rotasi pada setiap tahapannya, seperti terlihat pada ilustrasi Gambar II.5 (Peryanto, Yudhana, & Umar, 2020).




Gambar II.5 Ilustrasi 5-*Fold Cross Validation*

Sumber: (Peryanto et al., 2020)

Dalam *cross validation*, ada empat nilai yang umum digunakan untuk mengevaluasi kinerja sebuah model, yaitu *precision, recall*, *F-score,* dan *accuracy*. *Precision* mengukur jumlah instance yang tepat dikenali sebagai *positive* dibagi dengan semua instance yang dikenali (persamaan 2.1); *recall* mengukur jumlah instance tepat dikenali sebagai *positive* dibagi dengan semua instance yang bernilai *positive* (persamaan 2.2); *F-score* merupakan rata-rata terbobot dari nilai *precision* dan *recall,* bergantung pada fungsi bobot β (persamaan 2.3); *accuracy* merupakan rata-rata terbobot dari nilai *precision* dan *inverse precision* (persamaan 2.4) (Dalianis, 2018).

Presisi adalah rasio prediksi positif yang benar terhadap keseluruhan hasil prediksi positif. Presisi dihitung menggunakan Persamaan (2.1).

$$Precision = \frac{tp}{tp+fp} \quad \text{.....(2.1)}$$

*Recall* itu adalah rasio antara prediksi positif dan data global positif. Recall dihitung menggunakan Persamaan (2.2).

$$Recall = \frac{tp}{tp+fn} \quad \text{.....(2.2)}$$

F1-score adalah semacam keseimbangan antara akurasi dan daya ingat dalam sistem. Ini adalah rata-rata yang diselaraskan dari nilai presisi dan pemulihan. F1-score dihitung menggunakan Persamaan (2.3).

$$Fscore = (1+\beta^2)\ \frac{Precision*Recall}{\beta^2 * Precision+Recall} \quad \text{.....(2.3)}$$

Akurasi adalah rasio prediksi yang benar terhadap jumlah perkiraan agregat. Rumus untuk menghitung akurasi ditunjukkan pada Persamaan (2.4).

$$Accuracy = \frac{tp+tn}{tp+tn+fp+fn} \quad \text{..... (2.4)}$$

Mengenai evaluasi hasil yang digunakan, salah satunya adalah matriks konfusi yang diperoleh dari hasil akurasi, presisi, dan recall serta dari kurva ROC untuk mengukur nilai AUC. Dengan begitu, semakin besar *area under the curve* (AUC) maka hasil prediksi akan semakin baik. Berikut adalah Tabel III.4 dari matriks konfusi (Putri, 2021) :

Tabel II.4. *Confusion Matrix*

| **Actually** | **Prediction** | |
|---|---|---|
| | **True** | **False** |
| True | TP | FN |
| False | FP | TN |