

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian mengenai komparasi fungsi aktivasi ReLu, Tanh dan Sigmoid pada dataset CIFAR-10, CORA, 9 gas, ZINC, dan BipedalWalker-v2 menunjukkan bahwa fungsi aktivasi Tanh memiliki nilai *training time* paling rendah dibandingkan dengan fungsi aktivasi lainnya (Yuen et al., 2021).

Penelitian mengenai komparasi fungsi aktivasi ReLu, Tanh dan Sigmoid pada dataset *solar radiation time series* menunjukkan bahwa fungsi aktivasi ReLu memiliki nilai RMSE paling rendah dibandingkan dengan fungsi aktivasi lainnya (Peñalva et al., 2022).

Penelitian mengenai komparasi fungsi aktivasi ReLu, Tanh dan Sigmoid pada dataset dari Directorate of Agriculture, Gandhinagar dan Agro-meteorology department, Anand Agricultural University menunjukkan bahwa fungsi aktivasi Tanh memiliki nilai MAPE paling rendah dibandingkan dengan fungsi aktivasi lainnya (Bhojani & Bhatt, 2021).

Penelitian mengenai komparasi fungsi aktivasi ReLu, Tanh dan Sigmoid pada dataset MINST menunjukkan bahwa fungsi aktivasi Tanh memiliki nilai RMSE paling rendah dibandingkan dengan fungsi aktivasi lainnya (Zhang et al., 2018).

Penelitian mengenai komparasi fungsi aktivasi ReLu, Tanh dan Sigmoid pada dataset WDBC menunjukkan bahwa fungsi aktivasi ReLu memiliki nilai

akurasi paling tinggi dibandingkan dengan fungsi aktivasi lainnya (Al-shargabi et al., 2019).

Tabel II.1 Kinerja Fungsi Aktivasi ReLu, Tanh dan Sigmoid

Penulis	Metode	Hasil
(Yuen et al., 2021)	Dataset: CIFAR-10, CORA, 9 gas, ZINC, dan BipedalWalker-v2 Fungsi Optimasi: Adam Evaluasi: <i>training time</i> (TT)	TT ReLu = 2223,4 detik
		TT Tanh = 2071,6 detik
		TT Sigmoid = 2344,2 detik
(Peñalva et al., 2022)	Dataset: <i>solar radiation time series</i> dari <i>website</i> NASA Fungsi Optimasi: SGD, Adam, RMSprop Evaluasi: RMSE	RMSE ReLu = 0,13918
		RMSE Tanh = 0,14849
		RMSE Sigmoid = 0,14892
(Bhojani & Bhatt, 2021)	Dataset: data primer dari Directorate of Agriculture, Gandhinagar dan Agro-meteorology department, Anand Agricultural University Fungsi Optimasi: Adam Evaluasi: MAPE	MAPE ReLu = 0,09146
		MAPE Tanh = 0,07886
		MAPE Sigmoid = 0,24005
(Zhang et al., 2018)	Dataset: MINST Fungsi Optimasi: CROWN-Ada Evaluasi: RMSE	RMSE ReLu = 1,17882
		RMSE Tanh = 0,92292
		RMSE Sigmoid = 1,1361
(Al-shargabi et al., 2019)	Dataset: WDBC Fungsi Optimasi: Adam, RMSprop Evaluasi: RMSE	Akurasi ReLu = 96.74%
		Akurasi Tanh = 96.65%
		Akurasi Sigmoid = 96.73%

Penelitian mengenai kinerja fungsi aktivasi ReLu berdasarkan jumlah *hidden layer* dan *neuron* per *hidden layer* pada dataset citra *x-ray* pasien COVID-19, menunjukkan bahwa fungsi aktivasi ReLu memiliki nilai akurasi tertinggi pada arsitektur dengan 2 *hidden layer* dan 4 *neuron* per *hidden layer* (Ahsan et al., 2020).

Penelitian mengenai kinerja fungsi aktivasi ReLu berdasarkan jumlah *hidden layer* dan *neuron* per *hidden layer* pada dataset cuplikan kamera pada jalan tol Jakarta-Cikampek, menunjukkan bahwa fungsi aktivasi ReLu memiliki nilai RMSE terendah pada arsitektur dengan 3 *hidden layer* dan 32 *neuron* per *hidden layer* (Sunindyo & Satria, 2020).

Penelitian mengenai kinerja fungsi aktivasi ReLu berdasarkan jumlah *hidden layer* dan *neuron per hidden layer* pada dataset Corpus, menunjukkan bahwa fungsi aktivasi ReLu memiliki nilai akurasi tertinggi pada arsitektur dengan 2 *hidden layer* dan 128 *neuron per hidden layer* (Munasatya & Novianto, 2020).

Penelitian mengenai kinerja fungsi aktivasi ReLu berdasarkan jumlah *hidden layer* dan *neuron per hidden layer* pada dataset MNIST, menunjukkan bahwa fungsi aktivasi ReLu memiliki nilai akurasi tertinggi pada arsitektur dengan 3 *hidden layer* dan 1024 *neuron per hidden layer* (Nguyen et al., 2021).

Penelitian mengenai kinerja fungsi aktivasi ReLu berdasarkan jumlah *hidden layer* dan *neuron per hidden layer* menggunakan 2000 data *simulate inputs*, menunjukkan bahwa fungsi aktivasi ReLu memiliki nilai MSE terendah pada arsitektur dengan 2 *hidden layer* dan 8 *neuron per hidden layer* (Rynkiewicz, 2019).

Tabel II.2 Kinerja ReLu Berdasarkan Hidden Layer

Penulis	Metode	Parameter	Hasil
(Ahsan et al., 2020)	Dataset: data citra <i>x-ray</i> pasien COVID-19 dari repositori GitHub <i>Hidden layer</i> (HN): 1, 2, dan 3 Jumlah <i>neuron</i> (N): 4 dan 8 Evaluasi: Akurasi	HL = 1, N = 4	Akurasi = 42%
		HL = 1, N = 8	Akurasi = 56%
		HL = 2, N = 4	Akurasi = 100%
		HL = 2, N = 8	Akurasi = 35%
		HL = 3, N = 4	Akurasi = 33%
		HL = 3, N = 8	Akurasi = 64%
(Sunindyo & Satria, 2020)	Dataset: data cuplikan kamera pada jalan tol Jakarta-Cikampek, km. 07 <i>Hidden layer</i> (HL): 3 Jumlah <i>neuron</i> (N): 16, 32, dan 64 Evaluasi: RMSE	HL = 3, N = 16	RMSE = 2.83013
		HL = 3, N = 32	RMSE = 1.88875
		HL = 3, N = 64	RMSE = 2.9038
(Munasatya & Novianto, 2020)	Dataset: Corpus, sebanyak 3686 data <i>Hidden layer</i> (HL): 2 Jumlah <i>neuron</i> (N): 32, 64, dan 128 Evaluasi: RMSE	HL = 2, N = 32	Akurasi = 92.78%
		HL = 2, N = 64	Akurasi = 92.93%
		HL = 2, N = 128	Akurasi = 93.01%

(Nguyen et al., 2021)	Dataset: MNIST <i>Hidden layer</i> (HL): 32, 64, dan 128 Jumlah <i>neuron</i> (N): 512 dan 1024 Evaluasi: Akurasi	HL = 3, N = 512	Akurasi = 94.5%
		HL = 3, N = 1024	Akurasi = 96.1%
(Rynkiewicz, 2019)	Dataset: data <i>simulate inputs</i> sebanyak 2000 <i>Hidden layer</i> (HL): 2 Jumlah <i>neuron</i> (N): 8, 16, 32, 64 dan 128 Evaluasi: MSE	HL = 2, N = 8	MSE = 1.31
		HL = 2, N = 16	MSE = 1.67
		HL = 2, N = 32	MSE = 2.62
		HL = 2, N = 64	MSE = 2.16
		HL = 2, N = 128	MSE = 1.61

Penelitian mengenai kinerja fungsi aktivasi Tanh berdasarkan jumlah *hidden layer* dan *neuron* per *hidden layer* menggunakan sampel acak dari 111 data wajib militer, menunjukkan bahwa fungsi aktivasi Tanh memiliki nilai AUC tertinggi pada arsitektur dengan 18 *hidden layer* dan 20 *neuron* per *hidden layer* (Bekesiene et al., 2021).

Penelitian mengenai kinerja fungsi aktivasi Tanh berdasarkan jumlah *hidden layer* dan *neuron* per *hidden layer* menggunakan data dari enam batch fermentasi (A sampai F), menunjukkan bahwa fungsi aktivasi Tanh memiliki nilai akurasi tertinggi pada arsitektur dengan 1 *hidden layer* dan 10 *neuron* per *hidden layer* (Mamat et al., 2020).

Penelitian mengenai kinerja fungsi aktivasi Tanh berdasarkan jumlah *hidden layer* dan *neuron* per *hidden layer* menggunakan data sampel *tapping* jaringan Universitas Ibnu Sina Batam, menunjukkan bahwa fungsi aktivasi Tanh memiliki nilai *loss* terendah pada arsitektur dengan 1 *hidden layer* dan 14 *neuron* per *hidden layer* (I. Gunawan, 2020).

Penelitian mengenai kinerja fungsi aktivasi Tanh berdasarkan jumlah *hidden layer* dan *neuron* per *hidden layer* dengan variasi 10 dan 40 *neuron* per

hidden layer, menunjukkan bahwa fungsi aktivasi Tanh memiliki nilai *error* terendah pada arsitektur dengan 1 *hidden layer* dan 40 *neuron* per *hidden layer* (Karlik & Olgac, 2019).

Penelitian mengenai kinerja fungsi aktivasi Tanh berdasarkan jumlah *hidden layer* dan *neuron* per *hidden layer* menggunakan data *real* dari harga beras periode 1 Januari 2016 sampai 31 Januari 2020, menunjukkan bahwa fungsi aktivasi Tanh memiliki nilai RMSE terendah pada arsitektur dengan 1 *hidden layer* dan 14 *neuron* per *hidden layer* (Sen et al., 2020).

Tabel II.3 Kinerja Tanh Berdasarkan Hidden Layer

Penulis	Metode	Parameter	Hasil
(Bekesiene et al., 2021)	Dataset: sampel acak dari 111 data wajib militer di bulan-bulan pertama bertugas <i>Hidden layer</i> (HL): 18, 24, dan 31 Jumlah <i>neuron</i> (N): 20 Evaluasi: AUC	HL = 31, N = 20	AUC = 78.9%
		HL = 18, N = 20	AUC = 99.73%
		HL = 24, N = 20	AUC = 83.23%
(Mamat et al., 2020)	Dataset: data dari enam batch fermentasi (A sampai F) dicatat, dengan variasi waktu fermentasi 21 hingga 45 jam <i>Hidden layer</i> (HL): 1 Jumlah <i>neuron</i> (N): 3, 5, 9, 10 Evaluasi: Akurasi	HL = 1, N = 3	Akurasi = 90.4%
		HL = 1, N = 5	Akurasi = 90%
		HL = 1, N = 9	Akurasi = 92.2%
		HL = 1, N = 10	Akurasi = 94.9%
(I. Gunawan, 2020)	Dataset: data sampel <i>tapping</i> jaringan Universitas Ibnu Sina Batam selama 15 menit <i>Hidden layer</i> (HL): 1, 2, 3 Jumlah <i>neuron</i> (N): 14 Evaluasi: <i>loss</i>	HL = 1, N = 14	<i>Loss</i> = 0.35424
		HL = 2, N = 14	<i>Loss</i> = 0.37394
		HL = 3, N = 14	<i>Loss</i> = 0.40234
(Karlik & Olgac, 2019)	<i>Hidden layer</i> (HL): 1 Jumlah <i>neuron</i> (N): 10 dan 40 Evaluasi: <i>error</i>	HL = 1, N = 10	<i>Error</i> = 0.025
		HL = 1, N = 40	<i>Error</i> = 0.002

(Sen et al., 2020)	Dataset: data <i>real</i> harga beras dari 1 Januari 2016 sampai 31 Januari 2020 <i>Hidden layer</i> (HL): 3 Jumlah <i>neuron</i> (N): 64, 128 dan 256 Evaluasi: RMSE	HL = 1, N = 14	RMSE = 0.3683
		HL = 2, N = 14	RMSE = 0.6735
		HL = 3, N = 14	RMSE = 0.6992

Penelitian mengenai kinerja fungsi aktivasi *sigmoid* berdasarkan jumlah *hidden layer* dan *neuron* per *hidden layer* menggunakan data dari tiga stasiun pengukur aliran sungai di Rio do Carmo Basin, menunjukkan bahwa fungsi aktivasi *sigmoid* memiliki nilai RMSE terendah pada arsitektur dengan 1 *hidden layer* dan 2 *neuron* per *hidden layer* (de Souza et al., 2020).

Penelitian mengenai kinerja fungsi aktivasi *sigmoid* berdasarkan jumlah *hidden layer* dan *neuron* per *hidden layer* menggunakan data rekaman model sinyal ultrasonik M500, menunjukkan bahwa fungsi aktivasi *sigmoid* memiliki nilai akurasi tertinggi pada arsitektur dengan 4 *hidden layer* dan 16 *neuron* per *hidden layer* (Neto et al., 2021).

Penelitian mengenai kinerja fungsi aktivasi *sigmoid* berdasarkan jumlah *hidden layer* dan *neuron* per *hidden layer* menggunakan data 10000 citra FPGA, menunjukkan bahwa fungsi aktivasi *sigmoid* memiliki nilai akurasi tertinggi pada arsitektur dengan 3 *hidden layer* dan 10 *neuron* per *hidden layer* (Si et al., 2020).

Penelitian mengenai kinerja fungsi aktivasi *sigmoid* berdasarkan jumlah *hidden layer* dan *neuron* per *hidden layer* menggunakan data jumlah tersangka penyalahgunaan narkoba di Provinsi Bali tahun 2007-2018, menunjukkan bahwa fungsi aktivasi *sigmoid* memiliki nilai MAPE terendah pada arsitektur dengan 2 *hidden layer* dan 3 *neuron* pada *hidden layer* pertama dan 2 *neuron* pada *hidden layer* kedua (Githa Pratiwi et al., 2019).

Penelitian mengenai kinerja fungsi aktivasi *sigmoid* berdasarkan jumlah *hidden layer* dan *neuron* per *hidden layer* menggunakan data volume ekspor dan impor minyak dan gas bumi di indonesia, menunjukkan bahwa fungsi aktivasi *sigmoid* memiliki nilai akurasi terendah pada arsitektur dengan 1 *hidden layer* dan 5 *neuron* per *hidden layer* dan arsitektur dengan 1 *hidden layer* dan 10 *neuron* per *hidden layer* (Wanto et al., 2019).

Tabel II.4 Kinerja Sigmoid Berdasarkan Hidden Layer

Penulis	Metode	Parameter	Hasil
(de Souza et al., 2020)	Dataset: data dari tiga stasiun pengukur aliran sungai di Rio do Carmo Basin <i>Hidden layer</i> (HL): 1 Jumlah <i>neuron</i> (N): 1, 2, 3, dan 5 Evaluasi: RMSE	HL = 1, N = 2	RMSE = 1.697
		HL = 1, N = 5	RMSE = 2.3371
		HL = 1, N = 1	RMSE = 2.4057
		HL = 1, N = 3	RMSE = 2.1982
(Neto et al., 2021)	Dataset: data rekaman model sinyal ultrasonik M500 dari Petterson <i>Hidden layer</i> (HL): 1, 2, dan 4 Jumlah <i>neuron</i> (N): 6, 8, 12, dan 16 Evaluasi: Akurasi	HL = 1, N = 6	Akurasi = 81.92%
		HL = 1, N = 8	Akurasi = 81.92%
		HL = 1, N = 12	Akurasi = 81.92%
		HL = 1, N = 16	Akurasi = 81.92%
		HL = 2, N = 16	Akurasi = 82.17%
		HL = 4, N = 16	Akurasi = 82%
(Si et al., 2020)	Dataset: data 10000 citra FPGA <i>Hidden layer</i> (HL): 2 dan 3 Jumlah <i>neuron</i> (N): 10 Evaluasi: Akurasi	HL = 2, N = 10	Akurasi = 90.2%
		HL = 3, N = 10	Akurasi = 95.5%
(Githa Pratiwi et al., 2019)	Dataset: data jumlah tersangka penyalahgunaan narkoba di Provinsi Bali tahun 2007-2018 <i>Hidden layer</i> (HL): 2 Jumlah <i>neuron</i> (N): 2 dan 3 Evaluasi: MAPE	HL = 2, N = 2 & 2	MAPE = 5.2%
		HL = 2, N = 3 & 2	MAPE = 3.7%
(Wanto et al., 2019)	Dataset: data volume ekspor dan impor minyak dan gas bumi di indonesia, dari BPS <i>Hidden layer</i> (HL): 1 Jumlah <i>neuron</i> (N): 5, 8, 10, dan 14 Evaluasi: Akurasi	HL = 1, N = 5	Akurasi = 83%
		HL = 1, N = 8	Akurasi = 50%
		HL = 1, N = 10	Akurasi = 83%
		HL = 1, N = 14	Akurasi = 67%

2.2 Kinerja Pegawai

Pegawai merupakan salah satu bagian yang penting dalam perusahaan atau organisasi, di mana merekalah yang menjadi perencana, pelaksana dan pengendali yang memberikan peran aktif dalam mewujudkan tercapainya tujuan perusahaan atau organisasi tersebut. Kinerja pegawai merupakan hasil kerja yang dapat di capai oleh seseorang pegawai dalam suatu perusahaan atau organisasi, dalam melakukan suatu kegiatan pekerjaan, sesuai dengan tanggung jawab dan tujuan pekerjaan yang diharapkan. Kinerja pegawai dipengaruhi oleh beberapa faktor, seperti suasana kerja, kepemimpinan, kesejahteraan, rekan kerja dan hal-hal lain yang dapat mempengaruhi kemampuan pegawai tersebut dalam menjalankan tugasnya (Megawe et al., 2020). Kinerja pegawai merupakan faktor yang penting dalam sebuah perusahaan atau organisasi, karena kinerja pegawai merupakan nilai ukur efektivitas dari pencapaian tujuan perusahaan atau organisasi tersebut, dan dapat digunakan untuk mencerminkan keberhasilan para manajer dalam pengelolaan organisasi dan SDM (Eliana et al., 2020). Untuk mengukur tingkat pencapaian seorang pegawai terhadap setiap tujuan kinerja yang sudah ditetapkan sebelumnya, diperlukan deskripsi dan analisis yang tepat terhadap pencapaian kinerja tersebut (Hasanudin & Budiharjo, 2021).

Dalam mengelola dan merencanakan kinerja pegawai di perusahaannya, ada beberapa langkah yang dapat diambil oleh perusahaan atau organisasi, antara lain (Wanasaputra & Dewi, 2017):

1. Identifikasi tujuan
2. Identifikasi produk / jasa

3. Penilaian sumber daya internal
4. Penilaian lingkungan eksternal
5. Penilaian keuntungan kompetitif
6. Implementasi strategi
7. Evaluasi manfaat

2.3 *Machine Learning*

Machine learning merupakan bagian dari *Artificial 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 pada dasarnya dapat dibagi menjadi 4 kategori, yaitu *Supervised Learning*, *Unsupervised Learning*, *Semi-supervised learning*, dan *Reinforcement Learning*. Dalam *machine learning*, data pada umumnya dibagi menjadi dua jenis, yaitu data *training* yang digunakan untuk melatih algoritma dalam menemukan model yang cocok dan data *testing* yang digunakan untuk menguji dan menentukan kinerja model yang diperoleh (Ridwan et al., 2020).

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 et al., 2018). Selain itu, sifat *machine learning* yang konstruktif di dalam prosesnya menghasilkan sebuah hasil prediksi dan analisis yang akurat (Kohsasih & Situmorang, 2022).

2.4 Klasifikasi

Klasifikasi merupakan sebuah proses pembelajaran oleh mesin (machine learning) pada sebuah model, dengan menggunakan sekumpulan data latih, dan hasilnya digunakan untuk memprediksi atau mengklasifikasi data uji (Kusuma et al., 2022). Proses pembelajaran dalam klasifikasi dapat dianalogikan seperti pengalaman manusia dalam mempelajari sesuatu hal yang baru, dimana manusia akan menyimpan pengalaman-pengalaman baru yang dipelajarinya di dalam memori otak dan dapat dikeluarkan sebagai pengetahuan untuk memahami sesuatu yang baru di kemudian hari (Azhari et al., 2021).

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 (M. Gunawan et al., 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 et al., 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 et al., 2019).

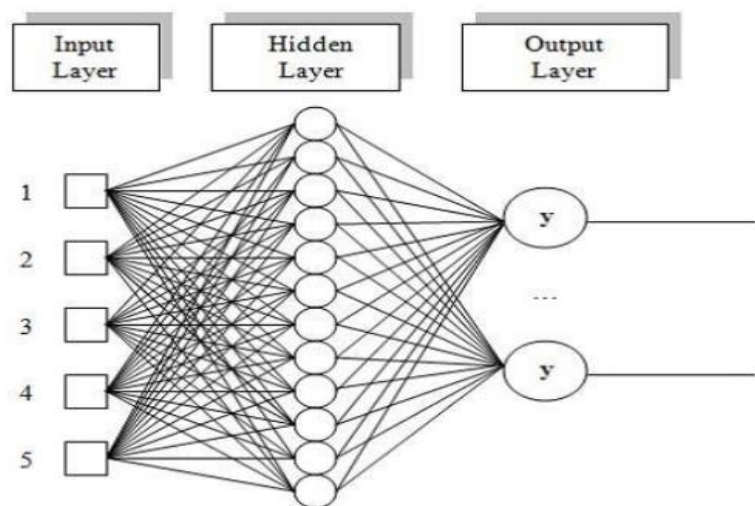
2.5 *Multilayer Perceptron*

Multilayer Perceptron (MLP) melakukan klasifikasi terhadap data dengan cara yang sama dengan perceptron, namun perbedaan keduanya terletak di jumlah hidden layer MLP yang dapat berjumlah lebih dari satu sehingga proses penghitungan bobot dan biasnya melalui lebih dari satu kali pemrosesan (Munasatya & Novianto, 2020).

MLP merupakan algoritma klasifikasi jenis *logistic regression*, di mana input pertama kali ditransformasi menggunakan *non-linear learning*, dengan tujuan untuk memproyeksikan data input ke dalam ruang yang dapat memisahkan data tersebut secara linier. Algoritma *backpropagation* merupakan algoritma yang paling umum digunakan dalam jaringan MLP, yang menggunakan prosedur *gradient descent*, dalam upayanya untuk mengurangi tingkat kesalahan antara hasil output jaringan dengan target yang ditentukan (Simanjuntak et al., 2020). Dalam prosesnya, algoritma ini melakukan perubahan terhadap nilai-nilai bobot dengan arah mundur (*backward*), dan kemudian melakukan proses *feed-forward*

untuk menghasilkan nilai *error*. Pada proses inilah neuron-neuron pada arsitektur MLP diaktifkan dengan menggunakan sebuah fungsi aktivasi (Pardede et al., 2022).

Karena merupakan algoritma yang bersifat *feed-forward*, yang terdiri dari sejumlah neuron yang dihubungkan oleh bobot-bobot penghubung, kinerja klasifikasi MLP sangat bergantung pada struktur jaringan yang dibentuk dalam proses pelatihannya. Lapisan dalam MLP terdiri dari lapisan awal yang memberi input kepada jaringan, dengan variabel masukan dilambangkan sebagai lapisan masukan (*input layer*), lapisan akhir yang disebut lapisan keluaran (*output layer*), dan semua lapisan diantara lapisan masukan dan keluaran yang disebut sebagai lapisan tersembunyi (*hidden layer*). Adapun bentuk umum struktur MLP seperti terlihat pada Gambar II.1 (Hardiyanti & Fitriana, 2021).



Gambar II.1 Bentuk Umum MLP
Sumber: (Hardiyanti & Fitriana, 2021)

Dalam proses pelatihannya, *backpropagation* secara umum melakukan tahapan-tahapan sebagai berikut (Saputra & Asdar, 2021):

1. Inisialisasi semua bobot dengan bilangan acak kecil.
2. Jika kondisi penghentian belum terpenuhi, lakukan langkah 3-9
3. Untuk setiap pasang data pelatihan, lakukan langkah 4-9
4. Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya

5. Hitung semua keluaran unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$z_net_j = v_{j0} + \sum_{i=1}^n (x_i v_{ij}) \dots\dots\dots \text{II.1}$$

$$z_j = f(z_net_j) = \frac{1}{1+e^{-z_net_j}} \dots\dots\dots \text{II.2}$$

6. Hitung semua keluaran jaringan di unit y_k ($k = 1, 2, \dots, m$)

$$y_net_k = w_{k0} + \sum_{j=1}^n (z_j w_{kj}) \dots\dots\dots \text{II.3}$$

$$y_k = f(y_net_k) = \frac{1}{1+e^{-y_net_k}} \dots\dots\dots \text{II.4}$$

7. Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k = 1, 2, \dots, m$)

$$\delta_k = (t_k - y_k) f'(y_net_k) = (t_k - y_k) y_k (1 - y_k) \dots\dots\dots \text{II.5}$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer dibawahnya. Hitung suku perubahan bobot w_{kj} dengan laju percepatan α .

$$\Delta w_{kj} = \alpha \delta_k z_j; \quad k = (1, 2, \dots, n) \dots\dots\dots \text{II.6}$$

8. Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$\delta_net_j = v_{j0} + \sum_{k=1}^n (\delta_k w_{kj}) \dots\dots\dots \text{II.7}$$

Faktor δ unit tersembunyi:

$$\delta_j = \delta_net_j f'(z_net_j) = \delta_net_j z_j (1 - z_j) \dots\dots\dots \text{II.8}$$

Hitung suku perubahan bobot v_{ij}

$$\Delta v_{kj} = \alpha \delta_i x_i; j = (1, 2, \dots, n) \dots\dots\dots \text{II.9}$$

9. Hitung semua perubahan bobot. Perubahan bobot garis yang menuju ke unit keluaran w_{kj} (baru)

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + v_{ji} \dots\dots\dots \text{II.10}$$

Dengan ($k = 1, 2, \dots, m$; $j = 0, 1, \dots, p$). Perubahan bobot garis yang menuju ke unit tersembunyi:

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \dots\dots\dots \text{II.11}$$

2.6 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*. Beberapa fungsi aktivasi yang umum digunakan di dalam *neural network* adalah fungsi ReLu, Tanh, dan Sigmoid (Euis Saraswati et al., 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 et al., 2020). ReLU memiliki kelebihan dalam hal kemampuannya membantu mengurangi biaya komputasi, mempercepat proses pelatihan, dan mencegah gradien yang hilang (Nguyen et al., 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):

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

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):

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

Fungsi Sigmoid merupakan fungsi aktivasi yang paling umum digunakan di dalam *neural network*, karena memiliki hasil akurasi yang cukup tinggi dan mudah untuk dibedakan dengan fungsi aktivasi yang lain. Dalam penggunaannya, fungsi Sigmoid dituliskan menggunakan persamaan (Si et al., 2020):

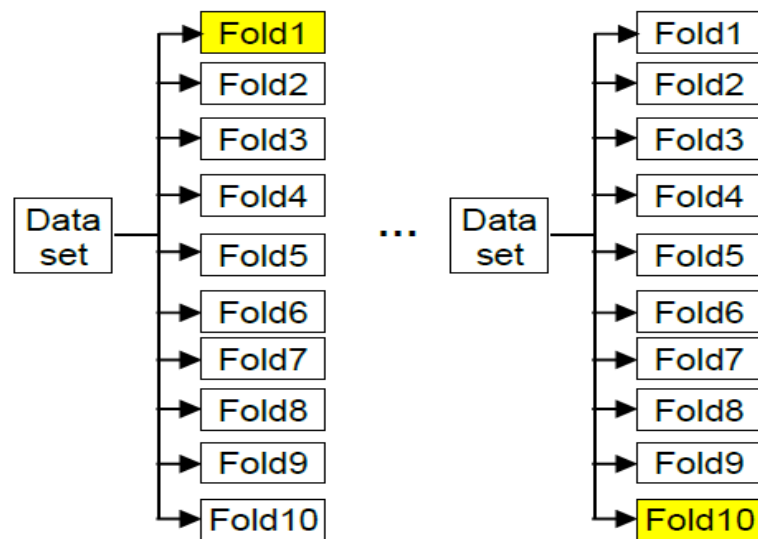
$$f(x) = \frac{1}{1 + e^{-x}} \dots\dots\dots \text{II.14}$$

2.7 Cross Validation

Cross validation merupakan teknik evaluasi yang biasanya digunakan untuk mengevaluasi sebuah model klasifikasi, dengan cara mengukur keakuratan atau tingkat kemiripan antara hasil klasifikasi dengan data yang sebenarnya (Lubis

et al., 2020). *Cross validation* membagi data ke dalam K bagian (*fold*), sehingga sering juga disebut dengan *k-fold cross validation* (Peryanto et al., 2020).

Sebagai contoh, jika digunakan nilai $K = 10$, maka teknik evaluasi yang digunakan adalah *10-fold cross validation*, dimana data dibagi sama rata ke dalam 10 bagian (*fold*). Dilakukan iterasi dari $I = 1$ sampai dengan $I = 10$ dengan mengambil data pada *fold* ke-I dan dilakukan evaluasi terhadap hasilnya. Gambar II.2 menunjukkan contoh dari pengambilan data pada evaluasi menggunakan *10-fold cross validation* (Handayani et al., 2022).



Gambar II.2 Contoh 10-fold Cross Validation

Sumber: (Handayani et al., 2022)

Evaluasi *cross validation* dapat digunakan bersamaan dengan evaluasi *confusion matrix*, untuk menghitung nilai-nilai seperti akurasi, presisi, *recall*, dan *f-measure*, berdasarkan nilai *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN) yang dihasilkannya (Furqan et al., 2022). Nilai-nilai ini pada umumnya disajikan dalam bentuk tabel dengan bentuk seperti terlihat pada Tabel II.5 (Sitepu et al., 2021).

Tabel II.5 *Confusion Matrix*

Kelas	Terklasifikasi Positif	Terklasifikasi Negatif
Positif	True Positive	False Positive
Negatif	False Negative	True Negative

Sumber: (Sitepu et al., 2021)

Untuk menghitung nilai *accuracy*, *precision*, *recall*, dan *f-measure*, dapat digunakan persamaan-persamaan berikut (Margolang et al., 2022):

$$Precision = \frac{tp}{tp+fp} \dots\dots\dots II.15$$

$$Recall = \frac{tp}{tp+fn} \dots\dots\dots II.16$$

$$Accuracy = \frac{tp+tn}{tp+tn+fp+fn} \dots\dots\dots II.17$$