

BAB II

LANDASAN TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menunjang proses penelitian. Materi utama yang dibahas dalam penelitian ini yaitu Hidroponik, Sistem Kendali, Algoritma *Artificial Neural Network* (ANN), Algoritma *Back Propagation* dan *Internet of Things* (IoT).

2.1 Hidroponik

2.1.1 Tanaman Hidroponik Selada



Gambar 2. 1 Hidroponik Selada (Yulita & Migusnawati, 2023)

Selada adalah salah satu tanaman yang sangat cocok untuk dibudidayakan secara hidroponik, terutama dengan sistem. Karakteristik selada yang berakar kecil dan ringan membuatnya ideal untuk metode ini, di mana larutan nutrisi mengalir tipis di atas akar dalam aliran yang terus menerus. Kondisi ini memberikan kelembaban, oksigen, dan nutrisi secara optimal, sehingga menunjang pertumbuhan selada yang lebih cepat dan lebih sehat dibandingkan metode konvensional. Hidroponik melakukan kontrol penuh atas unsur hara yang diterima selada, yang biasanya membutuhkan tingkat nutrisi dalam larutan pada kisaran pH 6–7 dan PPM (*Part Per Million*) antara 560–840, tergantung pada tahap pertumbuhannya. Dengan pemantauan rutin pada parameter tersebut, risiko

kekurangan atau kelebihan nutrisi bisa diminimalkan, yang berpengaruh pada kualitas daun dan hasil panen yang optimal. Karena tidak menggunakan tanah, sistem ini juga mengurangi risiko kontaminasi dari hama dan patogen yang umumnya terdapat di tanah, sehingga selada hidroponik dapat dipanen dalam kondisi yang lebih bersih dan lebih segar. (Zahra et al., 2023).

2.1.2 Pengaruh pH dan PPM AB Mix terhadap Hidroponik

Tabel pH dan PPM untuk Sayuran Daun		
Nama Sayuran	pH	PPM
Artichoke	6.5 - 7.5	560 - 1260
Asparagus	6.0 - 6.8	980 - 1200
Bawang Pre	6.5 - 7.0	980 - 1260
Bayam	6.0 - 7.0	1260 - 1610
Brokoli	6.0 - 6.8	1960 - 2450
Brussell Kecambah	6.5	1750 - 2100
Endive	5.5	1400 - 1680
Kailan	5.5 - 6.5	1050 - 1400
Kangkung	5.5 - 6.5	1050 - 1400
Kubis	6.5 - 7.0	1750 - 2100
Kubis Bunga	6.5 - 7.0	1750 - 2100
Pakcoy	7.0	1050 - 1400
Sawi Manis	5.5 - 6.5	1050 - 1400
Sawi Pahit	6.0 - 6.5	840 - 1680
Seledri	6.5	1260 - 1680
Selada	6.0 - 7.0	560 - 840
Silverbeet	6.0 - 7.0	1260 - 1610

Gambar 2. 2 Tabel pH dan PPM Sayuran Daun (Bayu Wn, 2016)

Pada sistem hidroponik, pH dan PPM larutan nutrisi AB Mix memiliki pengaruh besar terhadap pertumbuhan dan kesehatan tanaman. Nilai pH memengaruhi ketersediaan unsur hara bagi tanaman, karena setiap unsur hara memiliki kisaran pH tertentu di mana ia dapat diserap secara optimal oleh akar. Misalnya, pada nilai pH di kisaran 6–7, unsur hara seperti nitrogen, fosfor, dan kalium dapat diakses dengan baik oleh tanaman, termasuk selada. Jika pH larutan keluar dari kisaran optimal, beberapa unsur hara akan terikat dan sulit diserap oleh tanaman, sehingga tanaman bisa mengalami gejala kekurangan gizi atau stres. Untuk menjaga efisiensi dan keseimbangan nutrisi, pH Up (berbasis kalium hidroksida/KOH) atau pH Down (berbasis asam fosfat/ H_3PO_4) digunakan untuk

menyesuaikan pH larutan agar tetap berada dalam kisaran optimal. Pemantauan dan penyesuaian ini penting untuk memastikan tanaman memperoleh nutrisi yang optimal (Romalasari & Sobari, 2019).

Selain itu, menurut penelitian dari (Pohan & Oktoujournal, 2019) PPM (Part Per Million) menunjukkan konsentrasi nutrisi dalam larutan. Tanaman memerlukan konsentrasi tertentu untuk setiap tahap pertumbuhannya biasanya lebih rendah di tahap awal dan lebih tinggi saat tanaman tumbuh besar dan memerlukan lebih banyak energi. Misalnya, selada memerlukan PPM antara 560–840. Konsentrasi yang terlalu rendah dapat menyebabkan tanaman kekurangan nutrisi, menghambat pertumbuhan, atau mempengaruhi warna dan kualitas daun. Sebaliknya, konsentrasi yang terlalu tinggi dapat mengakibatkan penumpukan garam di akar yang menyebabkan "nutrient burn" atau terbakar akibat nutrisi berlebih, merusak jaringan akar, dan memperlambat pertumbuhan tanaman. Pada sistem ini, larutan nutrisi yang digunakan umumnya berupa campuran AB Mix, yang dirancang khusus untuk memenuhi kebutuhan tanaman hidroponik. AB Mix terdiri dari dua komponen utama, yaitu:

2.1.4.1. Nutrisi A

Mengandung kalsium nitrat ($\text{Ca}(\text{NO}_3)_2$) yang menyediakan kalsium (Ca) untuk kekuatan struktur tanaman dan nitrogen (N) untuk pertumbuhan vegetatif. Selain itu, komponen ini mengandung besi (Fe) dalam bentuk kelat seperti ferro EDDHA atau ferro DTPA, yang penting untuk pembentukan klorofil.

2.1.4.2. Nutrisi B

Berisi kalium fosfat (KH_2PO_4) yang memberikan fosfor (P) untuk energi metabolisme dan kalium (K) untuk pengaturan osmotik. Komponen ini juga

mencakup magnesium sulfat (MgSO_4) sebagai sumber magnesium (Mg) untuk fotosintesis dan sulfur (S) untuk sintesis protein, serta elemen mikro seperti mangan (Mn), seng (Zn), tembaga (Cu), dan molibdenum (Mo) yang mendukung berbagai fungsi enzimatik dan metabolisme tanaman.

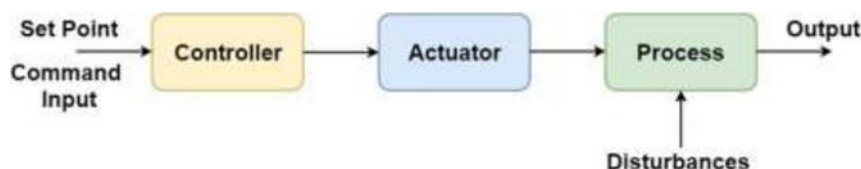
2.2 Sistem Kontrol

Suatu sistem didefinisikan sebagai sekumpulan komponen yang berinteraksi yang dirancang untuk mencapai tujuan bersama. Definisi ini berlaku di berbagai bidang, termasuk pendidikan dan proses industri (Control, 2015). Kontrol melibatkan pengukuran variabel dan menerapkan sinyal korektif untuk meminimalkan penyimpangan. Hal ini dapat dicapai melalui metode kontrol manual atau otomatis, di mana sistem otomatis menggunakan sensor dan aktuator untuk mempertahankan parameter yang diinginkan tanpa intervensi manusia (Goswami & Malhotra, 2017).

2.2.1 Sistem Kontrol Terbuka (*Open Loop*)

Sistem loop terbuka dicirikan oleh kurangnya mekanisme umpan balik, di mana transduser input mengubah input referensi menjadi variabel terkontrol tanpa menyesuaikan perubahan output. Struktur ini terbukti dalam berbagai aplikasi, termasuk terapi medis dan sistem pertanian. Transduser input memainkan peran penting dalam mengubah input referensi menjadi format yang dapat digunakan untuk pengontrol. Misalnya, dalam terapi ultrasound, transduser memberikan energi berdasarkan rencana yang telah ditentukan tanpa umpan balik. Pada perubahan suhu (Schönlein, 2021). Dalam sistem *loop* terbuka, input referensi diproses tanpa umpan balik, membuatnya rentan terhadap gangguan dari sinyal eksternal (Ahissar & Assa, 2016). Output, atau variabel terkontrol, adalah hasil

dari pemrosesan pengontrol dari input yang diubah, yang dapat dipengaruhi oleh penjumlahan persimpangan yang menggabungkan interferensi, penjumlahan ini dapat mempersulit respons sistem, karena tidak memperhitungkan mekanisme umpan balik yang dapat menstabilkan output (Buccioni et al., 2018).



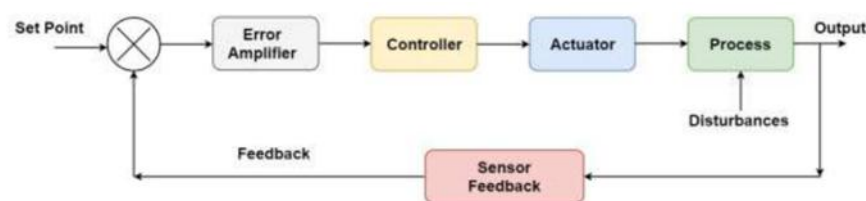
Gambar 2. 3 Sistem Kontrol Terbuka (Goswami & Malhotra, 2017)

Berdasarkan Gambar 2.3 kontrol *loop* terbuka sering lebih disukai dalam skenario di mana sumber daya komputasi terbatas, karena menyederhanakan arsitektur kontrol dengan menyiarkan sinyal identik ke beberapa *output*. Namun, tidak adanya umpan balik membuat sistem ini rentan terhadap gangguan, yang dapat menyebabkan penurunan kinerja, karena mereka tidak dapat menyesuaikan diri dengan perubahan lingkungan (Fieguth & Anglin, 2023). Sistem *loop* terbuka, ditandai dengan kurangnya umpan balik dari output ke tindakan kontrol, lazim dalam aplikasi seperti mesin cuci dan lampu lalu lintas. Kinerja mereka sangat bergantung pada kalibrasi, karena gangguan apa pun dapat menyebabkan penyimpangan yang signifikan dari hasil yang diinginkan (Wei et al., 2023).

2.2.2 Sistem Kontrol Tertutup (*Closed Loop*)

Dalam sistem *loop* tertutup, seperti yang ditunjukkan pada Gambar 2.4 interaksi antara transduser input dan output sangat penting untuk mempertahankan kinerja yang diinginkan. Transduser input memodifikasi input untuk pengontrol, sementara transduser *output* mengukur respons sistem, memasukkannya kembali ke pengontrol (Goswami & Malhotra, 2017). Persimpangan penjumlahan menggabungkan sinyal input dan *output*, menghasilkan sinyal *drive*, yang

mencerminkan kesalahan dalam sistem. Dalam sistem seperti konverter *flyback*, kontrol *loop* tertutup meningkatkan efisiensi dengan meminimalkan kerugian dibandingkan dengan konfigurasi *loop* buka (Umana Stuart & Kempf, n.d.). Gangguan diukur dan dimasukkan ke dalam jalur umpan balik, melakukan penyesuaian waktu nyata untuk menjaga stabilitas sistem. Teknik seperti analisis frekuensi *loop* tertutup memberikan wawasan tentang kinerja sistem dalam berbagai kondisi, meningkatkan ketahanan strategi kontrol (Dastjerdi et al., 2023).



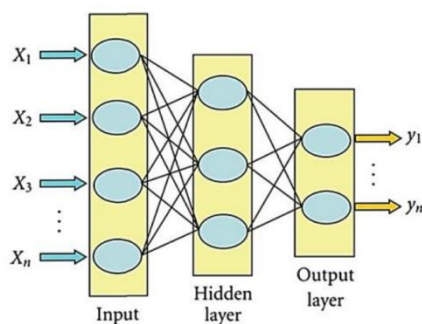
Gambar 2. 4 Sistem Kontrol Tertutup (Goswami & Malhotra, 2017)

2.3 Artificial Neural Network (ANN)

Artificial Neural Network (ANN), atau yang juga dikenal dengan istilah *Artificial Neural Network* (ANN), adalah suatu metode kecerdasan komputasi tingkat lanjut yang mampu menganalisis informasi dalam jumlah besar melalui proses pembelajaran dan pelatihan, mirip dengan cara kerja sistem kecerdasan manusia. JST merupakan salah satu metode yang sangat cocok untuk memodelkan hubungan antar unit yang kompleks, karena sifatnya yang non-linier, mampu memproses data secara paralel dengan banyak unit, serta dapat mentoleransi kesalahan yang terjadi. Meniru karakteristik neuron biologis, prinsip dasar proses ini adalah mengalikan setiap nilai input dengan faktor bobot tertentu, lalu menjumlahkan semua input yang telah terbobot untuk menentukan tingkat aktivasi neuron.

2.3.1 Arsitektur *Artificial Neural Network* (ANN)

Secara umum, arsitektur ANN terdiri dari tiga lapisan utama: lapisan input, lapisan tersembunyi, dan lapisan output. Setiap lapisan tersebut dapat memiliki jumlah *node* atau neuron yang bervariasi (Dananjaya et al., 2022). Seperti gambar berikut:



Gambar 2. 5 Lapisan ANN (Wijaya, 2019)

Berdasarkan Gambar 2.5 , arsitektur ANN terdiri dari:

- a. *Input layer*, yaitu lapisan neuron yang menerima sinyal dari luar untuk diteruskan ke neuron lain dalam jaringan, menyerupai fungsi sel saraf sensorik.
- b. *Hidden layer*, yaitu lapisan yang meniru fungsi sel saraf konektor pada jaringan biologis dan berperan meningkatkan kemampuan jaringan.
- c. *Output layer*, yaitu lapisan yang menyalurkan hasil sinyal keluaran dari proses jaringan, menyerupai fungsi sel saraf motorik pada jaringan biologis.

Menurut (Dananjaya et al., 2022) terdapat beberapa arsitektur ANN yang sering digunakan, salah satunya adalah:

1. Jaringan lapisan tunggal (*single layer network*)

Jaringan ini terdiri dari satu lapisan input dan satu lapisan *output*, di mana setiap neuron pada lapisan input terhubung langsung dengan neuron di lapisan

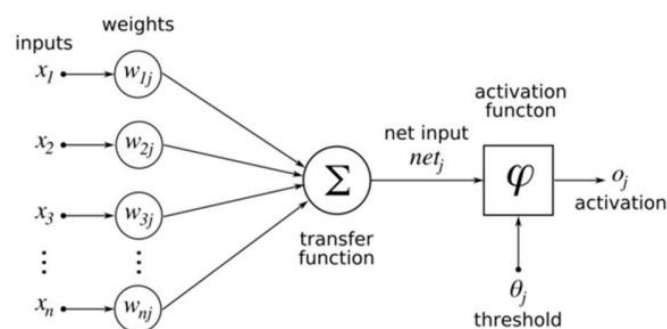
output. Sistem ini mengolah input langsung menjadi *output* tanpa melalui lapisan tersembunyi. Contoh algoritma ANN yang menggunakan metode ini adalah ADALINE, Hopfield, dan Perceptron.

2. Jaringan lapisan jamak (*multi layer network*)

Jaringan multilayer terdiri dari tiga jenis lapisan: *input layer*, *hidden layer*, dan *output layer*. Dibandingkan jaringan lapisan tunggal, jaringan *multilayer* mampu menangani masalah yang lebih kompleks. Namun, proses pelatihannya biasanya memakan waktu lebih lama. Contoh algoritma ANN yang menggunakan metode ini adalah *Madaline*, *backpropagation*, dan *neocognitron*.

2.3.2 Fungsi Aktivasi *Artificial Neural Network* (ANN)

Fungsi aktivasi dalam algoritma *backpropagation* berperan penting sebagai sinyal untuk menentukan keluaran yang akan diteruskan ke neuron lain. Peran fungsi ini sangat signifikan karena penggunaannya disesuaikan dengan kebutuhan dan tujuan tertentu. Fungsi aktivasi berpengaruh pada penyesuaian bobot dalam jaringan. Secara umum, fungsi ini menggambarkan hubungan antara tingkat aktivasi internal (*summation function*), yang dapat berbentuk linear maupun non-linear (Euis Saraswati et al., 2021).



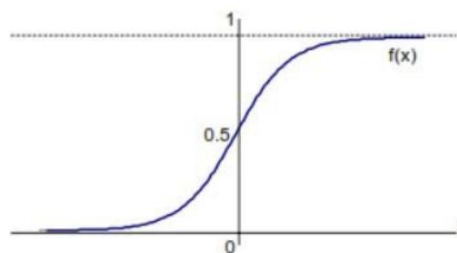
Gambar 2. 6 Struktur ANN (Euis Saraswati et al., 2021)

Pada struktur ANN yang ditunjukkan Gambar 2.6, fungsi transfer berperan

untuk mengubah sejumlah input (1, 2, 3...n) yang memiliki bobot (W_{1j} , W_{2j} , ... W_{nj}) dan bias menjadi nilai *input* (net_j) untuk fungsi aktivasi. Kemudian, fungsi aktivasi memproses nilai input tersebut dan membandingkannya dengan ambang batas (threshold) yang telah ditentukan, untuk menghasilkan *output* (o_j) setelah aktivasi dilakukan. Fungsi aktivasi menentukan apakah sinyal dari neuron input akan diteruskan atau dihentikan. Dengan demikian, fungsi ini mengatur apakah neuron akan "aktif" berdasarkan weighted sum dari input. Input tersebut kemudian diproses melalui fungsi propagasi. Dalam algoritma *backpropagation*, fungsi aktivasi harus memiliki karakteristik tertentu, seperti kontinu, dapat didiferensiasi, dan tidak meningkat secara monoton (Euis Saraswati et al., 2021).

Menurut (Kurniasari et al., 2023) Keefektifan komputasi ditentukan oleh kemudahan perhitungan turunan dari fungsi aktivasi tersebut. Beberapa fungsi aktivasi yang sering digunakan dalam algoritma *backpropagation* meliputi berbagai jenis sesuai dengan kebutuhan sistem.

1. Fungsi Aktivasi Sigmoid



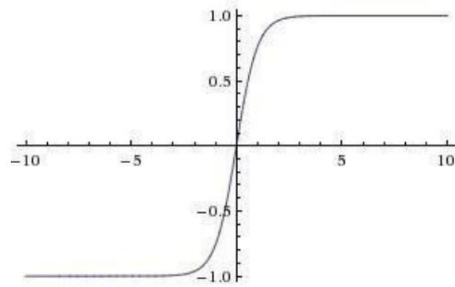
Gambar 2. 7 Fungsi Aktivasi Sigmoid

Pada Gambar 2.7 fungsi Sigmoid adalah fungsi non-linear dengan rentang nilai antara 0 hingga 1, sehingga sering digunakan dalam algoritma *backpropagation* yang membutuhkan input berupa bilangan real dan menghasilkan output dalam interval tersebut. Namun, fungsi ini memiliki kelemahan, seperti

potensi terjadinya masalah vanishing gradient dan output yang tidak berpusat pada nol. Berikut adalah bentuk fungsi Sigmoid.

$$F(x) = \frac{1}{1+e^{-x}} \dots\dots\dots(2.1)$$

2. Fungsi Aktivasi TanH

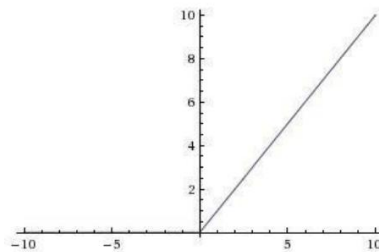


Gambar 2. 8 Fungsi Aktivasi TanH

Fungsi aktivasi Tanh adalah fungsi non-linear yang mengonversi nilai *zero-centered* input menjadi nilai dengan rentang antara -1 hingga 1, seperti yang ditunjukkan pada Gambar 2.8. Fungsi ini memiliki kelebihan berupa output yang bersifat *zero-centered*, namun juga memiliki kelemahan yaitu risiko terjadinya vanishing gradient. Fungsi Tanh merupakan bentuk pengembangan dari fungsi Sigmoid.

$$F(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1 \dots\dots\dots(2.2)$$

3. Fungsi Aktivasi *Rectified Linear Unit* (ReLU)

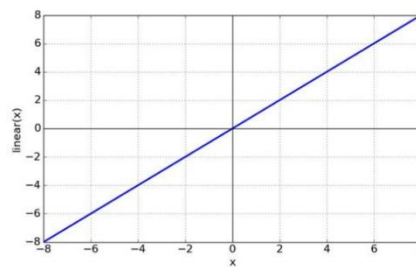


Gambar 2. 9 Fungsi Aktivasi ReLU

Berdasarkan Gambar 2.9 fungsi aktivasi ReLU adalah lapisan pada model

yang menerapkan fungsi $f(x)=\max(0,x)$. Fungsi ini pada dasarnya menetapkan batas pada nilai nol, sehingga jika $x \leq 0$, maka $x=0$, dan jika $x > 0$, maka x tetap sama. Keunggulan ReLU terletak pada kemampuannya mempercepat proses konvergensi yang dilakukan oleh *stochastic gradient descent* (SGD), dibandingkan dengan fungsi Sigmoid dan Tanh.

4. Fungsi Aktivasi Linear



Gambar 2. 10 Fungsi Aktivasi Linear

Berdasarkan Gambar 2.10 fungsi aktivasi linear adalah fungsi yang menghasilkan output yang sama dengan input, dan dapat dirumuskan sebagai berikut:

$$\text{Linear Function; } f(x) = x \dots\dots\dots (2.3)$$

Jika sebuah neuron menggunakan fungsi aktivasi linear, maka output yang dihasilkan adalah jumlah dari weighted sum input ditambah dengan bias.

2.3.3 Learning Rate

Learning rate adalah parameter dalam proses pelatihan yang digunakan untuk menghitung nilai koreksi bobot selama training. Nilai *learning rate* berkisar antara 0 hingga 1. Semakin tinggi nilai learning rate, proses pelatihan menjadi lebih cepat, tetapi akurasi jaringan cenderung menurun. Sebaliknya, jika *learning rate* lebih kecil, akurasi jaringan meningkat, namun proses pelatihan membutuhkan waktu yang lebih lama (Dananjaya et al., 2022).

2.3.4 Batch Size dan Optimizer

Batch size, atau ukuran *batch*, adalah istilah dalam pembelajaran mesin yang merujuk pada jumlah sampel pelatihan yang diproses dalam satu iterasi. *Hyperparameter* ini menentukan jumlah data yang digunakan sebelum model memperbarui parameter internalnya. Pada akhir setiap batch, prediksi dibandingkan dengan nilai output yang diharapkan, dan galat dihitung. Galat ini digunakan untuk memperbaiki model melalui algoritma pembaruan. Dataset pelatihan dapat dibagi menjadi beberapa batch, dengan ukuran yang sering digunakan adalah 32, 64, atau 128. *Optimizer*, atau algoritma optimasi, membantu menentukan bobot secara lebih efisien, mengurangi jumlah percobaan dan kesalahan saat proses *backpropagation* (Dananjaya et al., 2022).

2.3.5 Epoch dan Iterasi

Jumlah *epoch* adalah *hyperparameter* yang menentukan seberapa banyak algoritma pembelajaran akan memproses seluruh dataset pelatihan. Satu *epoch* berarti setiap sampel dalam dataset pelatihan digunakan sekali untuk memperbarui parameter model. Proses ini mencakup satu atau lebih *batch*. Dalam satu epoch, seluruh dataset dilatih pada jaringan neural hingga selesai dan diulang kembali dari awal untuk siklus berikutnya. Karena ukuran satu *epoch* biasanya terlalu besar untuk langsung dimasukkan ke dalam memori komputer, dataset perlu dibagi menjadi unit-unit kecil yang disebut batch. Sementara itu, iterasi adalah jumlah batch yang diperlukan untuk menyelesaikan satu *epoch*. Satu iterasi mencakup proses *forward pass* dan *backward pass* untuk memproses sebuah batch (Dananjaya et al., 2022).

2.3.6 Parameter Evaluasi *Artificial Neural Network* (ANN)

Untuk menilai akurasi hasil peramalan, dilakukan evaluasi dengan menggunakan berbagai teknik atau metode. Dalam penelitian ini, metode yang digunakan untuk mengevaluasi hasil peramalan adalah *Mean Squared Error* (MSE), Korelasi Pearson (R), *Mean Absolute Percentage Error* (MAPE) dan Standar Deviasi (SD) (Kurniasari et al., 2023).

1. *Mean Squared Error* (MSE)

Mean Squared Error (MSE) adalah salah satu metode yang digunakan untuk mengevaluasi akurasi metode peramalan. Dalam pendekatan ini, setiap kesalahan atau selisih antara nilai ramalan dan nilai aktual dikuadratkan. Proses pengkuadratan ini memberikan bobot lebih besar pada kesalahan yang besar, sehingga cenderung menonjolkan deviasi yang signifikan. Meskipun demikian, MSE menghasilkan evaluasi yang lebih baik untuk kesalahan kecil. MSE dihitung sebagai rata-rata dari kuadrat selisih antara nilai prediksi dan nilai yang diamati. Namun, kekurangannya adalah MSE dapat terlalu sensitif terhadap kesalahan besar karena sifat pengkuadratannya (Kurniasih et al., 2013). MSE dapat dihitung dengan rumus:

$$MSE = \sum_{t=1}^n \frac{(x_t - F_t)^2}{n} \dots \dots \dots (2.4)$$

Keterangan :

F = Nilai peramalan pada periode t

n = Jumlah data

x_t = Data Aktual

2. *Root Mean Squared Error* (RMSE)

Root Mean Squared Error (RMSE) adalah salah satu metrik standar yang

sering digunakan untuk mengukur tingkat kesalahan rata-rata dari suatu model prediksi atau peramalan. Metrik ini pada dasarnya adalah akar kuadrat dari *Mean Squared Error* (MSE). Dengan mengakarkuadratkan MSE, RMSE mengembalikan satuan kesalahan ke satuan asli dari variabel target, sehingga lebih mudah diinterpretasikan secara langsung dibandingkan MSE (Kurniasih et al., 2013).

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (x_t - F_t)^2} \dots \dots \dots (2.5)$$

Keterangan :

F_t = Nilai peramalan prediksi pada data ke- t

x_t = Nilai aktual pada data ke- t

n = Banyaknya data

3. Koefisien Korelasi Pearson

Korelasi adalah salah satu metode statistik yang digunakan untuk mengidentifikasi keberadaan hubungan serta menentukan arah hubungan antara dua atau lebih variabel. Sementara itu, koefisien korelasi merupakan angka yang menggambarkan seberapa kuat atau lemahnya hubungan tersebut. Koefisien korelasi sering dimanfaatkan sebagai metrik untuk mengukur tingkat kesamaan atau perbedaan antara dua vektor fitur (Mahsuri, 2023).

$$R = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{\sqrt{(n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2) (n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2)}} \dots (2.6)$$

Keterangan :

R = Nilai koefisien korelasi antara x dan y

n = Banyaknya data

x_i = Nilai x ke- i

y_i = Nilai y ke-i

Tabel 2. 1 Hubungan Koefisien Korelasi Pearson

Interval Koefisien	Tingkat Korelasi
0 – 0.20	Sangat lemah
0.21 - 0.40	Lemah
0.41 – 0.70	Sedang
0.71 – 0.90	Kuat
0.91 – 0.99	Sangat Kuat
1	Sempurna

4. *Standard Deviasi (SD)*

Standar deviasi adalah ukuran statistik yang digunakan untuk menunjukkan seberapa besar variasi atau penyebaran data dari rata-rata ($\bar{x}$). Semakin kecil nilai standar deviasi, semakin dekat data terhadap rata-rata, menandakan konsistensi atau stabilitas. Sebaliknya, standar deviasi yang besar menunjukkan data tersebar jauh dari rata-rata, yang berarti terdapat variasi atau fluktuasi yang signifikan (Putra & Cahyadi, 2016).

$$SD = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} \dots\dots\dots(2.7)$$

Keterangan:

x_i = Data ke-i

$\bar{x}$ = Rata-rata data

n = Jumlah data

5. *K Fold Cross Validation*

Cross-Validation (CV), yang juga dikenal sebagai estimasi rotasi, merupakan suatu teknik statistika inferensial fundamental yang digunakan untuk memvalidasi akurasi dan generalisasi sebuah model prediktif. Model ini dikembangkan berdasarkan dataset spesifik (Hafid, 2023). Pendekatan CV secara

efektif mengatasi isu-isu validasi dengan cara mempartisi data input menjadi beberapa subset (fold) yang independen. Dalam setiap iterasi CV, salah satu fold berfungsi sebagai set pengujian (testing set), sementara fold lainnya secara kolektif membentuk set pelatihan (training set). Proses implementasi CV melibatkan pembagian dataset awal menjadi sejumlah 'k' fold secara acak. Pada setiap siklus percobaan, satu fold dialokasikan sebagai set pengujian, dan 'k-1' fold sisanya digunakan sebagai set pelatihan. Prosedur ini diulang sebanyak 'k' kali, memastikan bahwa setiap fold mendapatkan kesempatan untuk berperan sebagai set pengujian, sehingga menghasilkan estimasi performa model yang lebih robust. Jumlah percobaan yang dilakukan secara langsung berbanding lurus dengan jumlah partisi (fold) yang telah ditentukan. Akurasi merupakan metrik evaluasi kuantitatif yang esensial untuk menilai kesesuaian hasil pengujian model dengan nilai sebenarnya. Nilai akurasi dapat diekspresikan dalam skala persentase (0% hingga 100%) atau dalam skala desimal (0 hingga 1). Semakin tinggi nilai akurasi yang diperoleh, semakin besar kapabilitas sistem dalam memprediksi atau mengklasifikasikan data secara tepat. Interpretasi spesifik mengenai rentang nilai akurasi suatu sistem biasanya dapat ditemukan pada Tabel 2.2 yang memberikan panduan kontekstual terhadap kinerja model.

Tabel 2. 2 Interpretasi Nilai Akurasi

Akurasi	Keterangan
< 0.25	Kurang Baik
0.25 – 0.5	Cukup Baik
0.5 – 0.75	Baik
> 0.75	Sangat Baik

2.4 Algoritma Back Propagation

Backpropagation adalah algoritma pembelajaran dalam *Artificial Neural Network* (ANN) yang sering digunakan untuk menyelesaikan masalah-masalah kompleks. Meskipun algoritma ini sudah ada sejak tahun 1970-an, namun mulai populer setelah diperkenalkan oleh David Rumelhart, Geoffrey Hinton, dan Ronald Williams dalam publikasi mereka pada tahun 1986. Dalam algoritma ini, terdapat dua tahap perhitungan: pertama, perhitungan maju untuk menghitung kesalahan antara output ANN dan target yang diinginkan, dan kedua, perhitungan mundur yang memanfaatkan kesalahan tersebut untuk memperbaiki bobot di seluruh neuron (Rubiani, 2019).

Menurut (Rubiani, 2019) terdapat beberapa fase dalam proses kerja *backpropagation*, yaitu:

1. Normalisasi Data

Normalisasi data bekerja dengan merubah skala nilai data asli ke dalam rentang tertentu berdasarkan nilai minimum dan maksimum dalam dataset. Pendekatan ini dilakukan dengan cara melakukan transformasi linier terhadap setiap nilai dalam dataset, sehingga nilai terkecil dalam dataset dipetakan menjadi batas bawah rentang yang diinginkan, sedangkan nilai terbesar dipetakan menjadi batas atasnya. Teknik ini berguna dalam berbagai aplikasi, terutama dalam pembelajaran mesin dan jaringan saraf tiruan, guna memastikan bahwa setiap fitur memiliki kontribusi yang seimbang dalam proses perhitungan.

Secara matematis, normalisasi *Min-Max* dinyatakan dengan persamaan berikut:

$$x'_n = a + \frac{x_n - x_{min}}{x_{max} - x_{min}} \cdot (b - a) \dots\dots\dots(2.8)$$

Keterangan:

x_n : Nilai asli dalam dataset

x_{min} : Nilai minimum dalam dataset

x_{max} : Nilai maksimum dalam dataset

a : Batas bawah rentang normalisasi

b : Batas atas rentang normalisasi

2. Inisialisasi bobot dan bias

Bobot dan bias dilakukan inisialisasi awal acak dengan nilai acak mendekati 0. Dalam penelitian ini inisialisasi bobot dan bias menggunakan pendekatan Nguyen Widrow agar distribusi bobot dan bias yang seimbang sehingga dapat mempercepat konvergensi. Pada Proses Nguyen Widrow inisialisasi nilai β digunakan untuk menyesuaikan skala bobot awal agar distribusinya lebih stabil.

$$\beta = 0.25 \times P_n^{\frac{1}{n}} \dots \dots \dots (2.9)$$

Keterangan:

0.25 : Faktor koreksi nguyen widrow

P : Jumlah *hidden layer*

n : Jumlah fitur input

Inisialisasi bobot dan bias pertama dilakukan dari input layer ke *hidden layer* dimana setiap bobot dinormalisasi menggunakan panjang vektor (norma) untuk memastikan distribusi nilai bobot awal v_{ij} tetap dalam rentang yang sesuai, dengan persamaan:

$$||v_{ij}|| = \sqrt{\sum_{i=1}^{n_{in}} v_{ij}^2} \dots \dots \dots (2.10)$$

Setelah vector (norma) didapat dan menghasilkan nilai acak maka setiap elemen dalam bobot v_{ij} diperbarui menggunakan rumus:

$$v_{ij} = \frac{\beta \times v_{ij}}{\|v_{ij}\|} \dots \dots \dots (2.11)$$

Setelah itu inialisasi bias awal menggunakan Nguyen Widrow dapat dituliskan menggunakan rumus sebagai berikut:

$$v_{0j} = 2 \times \beta \times \text{rand}(1, \text{hidden}) - \beta \dots \dots \dots (2.12)$$

Dimana

$\text{rand}(1, \text{hidden})$: merupakan nilai acak dalam rentang 0 sampai 1

Kemudian proses inialisasi bobot dari *hidden layer* dengan *Nguyen Widrow* dapat dituliskan sebagai berikut:

$$w_{jk} = \text{rand}(n_{\text{hidden}}, n_{\text{out}}) - 0,5 \dots \dots \dots (2.13)$$

Dimana $\text{rand}(n_{\text{hidden}}, n_{\text{out}})$ menghasilkan nilai acak antara 0 sampai 1 dikurangi 0,5 sehingga bobot antara *hidden layer* ke *output layer* berkisar antara [-0.5,0.5]

Kemudian proses inialisasi bias dari *hidden layer* ke *output layer* dengan *Nguyen Widrow* dapat dituliskan sebagai berikut:

$$w_{0k} = \text{rand}(1, n_{\text{out}}) - 0,5 \dots \dots \dots (2.14)$$

Dimana $\text{rand}(1, n_{\text{out}})$ menghasilkan nilai acak antara 0 sampai 1 dikurangi 0,5 sehingga bobot antara *hidden layer* ke *output layer* berkisar antara [-0.5,0.5].

3. *Feedforward*

Pada tahap ini, aliran data dilakukan secara maju seperti biasa dalam jaringan saraf tiruan. Semua nilai input beserta variabel yang ada di lapisan input dikirimkan ke lapisan tersembunyi (*hidden layer*). Nilai-nilai ini diterima oleh sejumlah *node* yang ada di lapisan tersembunyi, kemudian dihitung menggunakan

fungsi aktivasi. Hasil perhitungan dari setiap node di lapisan tersembunyi diberikan bobot dan diteruskan ke lapisan *output*. Pada lapisan *output*, dilakukan perhitungan lebih lanjut menggunakan fungsi aktivasi tertentu untuk menghasilkan output berdasarkan pola input yang diberikan.

Setiap input ($x_i, i = 1, 2, \dots, n$) menerima sinyal input x_i kemudian sinyal diteruskan pada *hidden layer* ke seluruh unit. Data training digunakan sebagai data input. Setiap hidden $Z_j, j = 1, 2, \dots, p$ akan menambahkan bobot dan bias sinyal input dengan persamaan berikut.

$$z_{inj} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots(2.15)$$

Dimana:

z_{inj} : keluaran *hidden layer*

x_i : Neuron input

V_{0j} : bias input *layer*

V_{ij} : bobot input *layer*

Dalam menghitung sinyal keluaran *hidden* unit menggunakan fungsi aktifitas *sigmoid*, kemudian mengirim sinyal *output* ke semua unit pada *output* dengan persamaan.

$$Z_j = f(Z_{inj}) \dots\dots\dots(2.15)$$

Tiap output ($y_k, k = 1, 2, \dots, m$) menambahkan bobot dan bias sinyal input dengan persamaan

$$y_{ink} = w_{ok} + \sum_{j=1}^n w_{jk} Z_j \dots\dots\dots(2.16)$$

Dimana:

y_{ink} : Sinyal keluaran *output layer*

w_{ok} : Bobot bias ke *output layer*

Z_j : Hasil fungsi aktivitas *hidden layer*

w_{jk} : Bobot *output layer*

Untuk menghitung sinyal keluaran dari *hidden unit* dengan fungsi aktivitas *purelin* atau identitas, kemudian mengirim sinyal *output* ke seluruh unit dalam unit *output* dengan persamaan

$$Y_k = f(y_{ink}) \dots\dots\dots(2.16)$$

4. Propagasi Kesalahan (*Backpropagation Error*)

Setelah jaringan saraf menghasilkan output dari perhitungan input seperti yang dijelaskan dalam tahap *feedforward*, langkah selanjutnya adalah membandingkan hasil yang diperoleh oleh setiap node *output* dengan nilai output yang diinginkan. Fungsi loss digunakan untuk menghitung kesalahan (*error*) pada setiap node. Kemudian, kesalahan ini dipropagasikan kembali mulai dari lapisan *output*, sehingga semua node, baik yang ada di lapisan output maupun tersembunyi, mendapatkan gambaran kesalahan terhadap output yang diinginkan. Setiap output ($Y_k, k = 1, 2, \dots, m$) menerima target pattern (desired output) yang sesuai dengan input training pattern guna menghitung kesalahan atau error tiap target dengan output yang didapatkan oleh jaringan dengan persamaan.

$$\delta_k = (t_k - Y_k)Y_k(1 - Y_k) \dots\dots\dots(2.17)$$

Keterangan:

δ_k : Faktor koreksi *error output layer*

t_k : Target

Y_k : Hasil *output training*

Menghitung perubahan bobot W_{jk} dengan persamaan

$$\Delta W_{jk} = \alpha \delta_k Z_j \dots\dots\dots(2.18)$$

Setelah iterasi pertama, maka dilakukan pembaruan bobot disertai momentum dengan persamaan

$$\Delta w_{jk(\text{baru})} = \alpha \cdot \delta_k \cdot z_j + (\mu \times \Delta w_{jk(\text{lama})}) \dots\dots\dots(2.19)$$

Keterangan:

ΔW_{jk} : Perubahan bobot *output layer*

α : *Learning rate*

δ_k : Faktor koreksi *error output layer*

Z_j : Hasil fungsi aktivitas *hidden layer*

μ : Momentum

$\Delta w_{jk(\text{lama})}$: Perubahan bobot sebelumnya

Menghitung perubahan bias W_{ok} dengan persamaan

$$\Delta W_{ok} = \alpha \delta_k \dots\dots\dots(2.20)$$

Setelah iterasi pertama dilakukan perubahan bias disertai momentum dengan persamaan

$$\Delta w_{ok(\text{baru})} = \alpha \cdot \delta_k + (\mu \times \Delta w_{ok(\text{lama})}) \dots\dots\dots(2.21)$$

Keterangan:

ΔW_{ok} : Perubahan bias *output layer*

α : *Learning rate*

δ_k : Faktor koreksi *error output layer*

μ : Momentum

$\Delta w_{ok(\text{lama})}$: Perubahan bias sebelumnya

Menghitung faktor koreksi *input hidden layer* dengan persamaan

$$\delta_{in j} = \delta_k w_{jk} \dots\dots\dots(2.22)$$

Keterangan:

δ_{inj} : Faktor koreksi *input hidden layer*

δ_k : Faktor koreksi *error output layer*

w_{jk} : Bobot *hidden layer* ke *output layer*

Menghitung faktor koreksi *error hidden unit* dengan persamaan

$$\delta_j = \delta_{inj}(1 - z_j^2) \dots\dots\dots(2.23)$$

Keterangan:

δ_j : Faktor koreksi *error hidden layer*

δ_{inj} : Faktor koreksi *input hidden layer*

z_j : Hasil fungsi aktivasi *hidden layer*

Menghitung koreksi bobot *hidden layer* dengan persamaan

$$\Delta v_{ij} = \alpha \delta_j x_j \dots\dots\dots(2.24)$$

Setelah iterasi pertama proses pelatihan *gradient descent* dengan momentum melakukan koreksi bobot *hidden layer* menggunakan persamaan

$$\Delta v_{ij(bar)} = \alpha \cdot \delta_j \cdot x_j + (\mu \times \Delta v_{ij(lama)}) \dots\dots\dots(2.25)$$

Keterangan:

Δv_{ij} : Perubahan bobot *hidden layer*

α : Learning rate

δ_j : Faktor koreksi *error hidden unit*

x_j : Nilai input

μ : Nilai momentum

$\Delta v_{ij(lama)}$: Perubahan bobot *hidden layer* sebelumnya

Menghitung koreksi bias *hidden layer* dengan persamaan

$$\Delta v_{oj} = \alpha \delta_j \dots\dots\dots(2.26)$$

Setelah iterasi pertama proses pembaruan bias hidden layer dengan gradient descent disertai momentum menggunakan persamaan

$$\Delta v_{0j}(\text{baru}) = \alpha \cdot \delta_j + (\mu \times \Delta v_{0j}(\text{lama})) \dots\dots\dots(2.27)$$

Keterangan:

Δv_{0j} : Perubahan bias *layer*

α : *Learning rate*

δ_j : Faktor koreksi *error hidden* unit

μ : Nilai momentum

$\Delta v_{0j}(\text{lama})$: Perubahan bias hidden layer sebelumnya

5. Pembaruan Bobot

Setelah kesalahan dihitung, nilai kesalahan ini digunakan untuk menghitung gradien dari fungsi loss dengan mempertimbangkan bobot pada jaringan. Gradien yang dihitung selanjutnya digunakan untuk mencari nilai bobot yang optimal, yang kemudian digunakan untuk memperbarui bobot pada jaringan saraf tersebut.

Perhitungan terhadap bobot baru *input layer* ke *hidden layer* dengan persamaan

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \dots\dots\dots(2.28)$$

Keterangan:

$v_{ij}(\text{baru})$: Bobot baru input layer ke *hidden layer*

$v_{ij}(\text{lama})$: Bobot lama input layer ke *hidden layer*

Δv_{ij} : Perubahan bobot *hidden layer*

Perhitungan terhadap bias baru *input layer* ke *hidden layer* dengan persamaan

$$v_{0j}(\text{baru}) = v_{0j}(\text{lama}) + \Delta v_{0j} \dots\dots\dots(2.29)$$

Keterangan:

$v_{0j}(\text{baru})$: Bias baru input layer ke *hidden layer*

$v_{0j}(\text{lama})$: Bias lama input layer ke *hidden layer*

Δv_{0j} : Perubahan bias layer

Perhitungan terhadap bobot baru *hidden layer* ke *output layer* dengan persamaan

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \dots\dots\dots(2.30)$$

Keterangan:

$w_{jk}(\text{baru})$: Bobot baru *hidden layer* ke *output layer*

$w_{jk}(\text{lama})$: Bobot lama *hidden layer* ke *output layer*

Δw_{jk} : Perubahan bobot *output layer*

Perhitungan terhadap bias baru *hidden layer* ke *output layer* dengan persamaan

$$w_{0k}(\text{baru}) = w_{0k}(\text{lama}) + \Delta w_{0k} \dots\dots\dots(2.31)$$

Keterangan:

$w_{0k}(\text{baru})$: Bias baru *hidden layer* ke *output layer*

$w_{0k}(\text{lama})$: Bias lama *hidden layer* ke *output layer*

Δw_{0k} : Perubahan bias *output layer*

2.5 Sensor

Sensor adalah perangkat keras atau perangkat (*device*) yang berfungsi untuk merespons rangsangan atau perubahan yang terjadi di lingkungan sekitar, seperti panas, cahaya, tekanan, suara, gerakan, dan lainnya. Sensor berperan sebagai penghubung input yang mendeteksi berbagai stimulus dari lingkungan.

Dalam penelitian ini, digunakan berbagai sensor gas serta sensor untuk mengukur suhu dan kelembaban.

2.5.1. Sensor PH



Gambar 2. 11 Sensor pH (Susilo et al., 2023)

Berdasarkan Gambar 2.11 sensor pH adalah perangkat yang digunakan untuk mengukur tingkat keasaman atau kebasaan suatu larutan berdasarkan konsentrasi ion hidrogen (H^+). Sensor ini bekerja dengan mendeteksi perubahan tegangan listrik yang dihasilkan oleh perbedaan konsentrasi ion H^+ antara elektroda kaca sensitif dan elektroda referensi. Nilai pH yang terukur berkisar dari 0 hingga 14, di mana $pH < 7$ menunjukkan sifat asam, $pH = 7$ netral, dan $pH > 7$ basa. Sensor pH umum digunakan dalam aplikasi lingkungan, industri, dan laboratorium untuk memantau kualitas air atau proses kimia (Susilo et al., 2023). Dalam penelitian ini digunakan sensor pH dengan spesifikasi sebagai berikut:

Tabel 2. 3 Spesifikasi Sensor PH

Fitur	Spesifikasi
Model	DFRobot Gravity Analog pH Sensor V1.1
Tegangan Operasi	3.3V - 5V DC
Output Tegangan	0 - 3.0V (analog)
Akurasi	± 0.1 pH (pada suhu standar)
Rentang Pengukuran	0 - 14 pH
Kompatibilitas	Arduino, Raspberry Pi, dan mikrokontroler
Dimensi Board	42mm x 32mm

Panjang Kabel Probe	100 cm
Suhu Operasi	0°C - 60°C
Resistansi Masukan	$\geq 100 \text{ G}\Omega$

2.5.2. Sensor PPM



Gambar 2. 12 Sensor PPM (Chuzaini & Dzulkiflih, 2022)

Sensor PPM air adalah perangkat yang digunakan untuk mengukur tingkat *Total Dissolved Solids* (TDS) atau konsentrasi zat terlarut dalam air dalam satuan *part per million* (ppm) seperti yang ditunjukkan Gambar 2.12. Sensor ini biasanya bekerja dengan mengukur konduktivitas listrik air, yang berkorelasi dengan jumlah ion terlarut. Sensor PPM air digunakan untuk memantau kualitas air dalam aplikasi seperti pengolahan air, akuakultur, dan penelitian lingkungan (Chuzaini & Dzulkiflih, 2022).

Tabel 2. 4 Spesifikasi Sensor PPM

Fitur	Spesifikasi
Model	DFRobot Gravity Analog TDS Sensor V1.0
Tegangan Operasi	3.3V - 5V DC
Output Tegangan	0 - 2.3V (analog)
Rentang Pengukuran	0 - 1000 ppm
Akurasi	$\pm 10\%$ (dengan larutan standar)
Suhu Operasi	0°C - 40°C
Kompatibilitas	Arduino, ESP32, dan mikrokontroler
Probe	Elektrode tahan korosi (kabel panjang 100 cm)
Dimensi Modul	42mm x 32mm

2.5.3. Sensor Suhu DS18B20



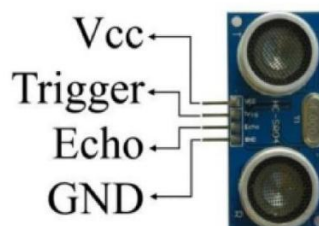
Gambar 2. 13 Sensor Suhu DS18B20

Sensor LM35 adalah sensor analog yang banyak digunakan dalam berbagai aplikasi elektronik, terutama untuk sistem pengukuran dan kontrol suhu. Berdasarkan Gambar 2.13 sensor ini tersedia dalam bentuk probe dan memiliki fitur tahan air, sehingga cocok untuk berbagai kondisi lingkungan. Sensor LM35 menghasilkan keluaran analog berupa tegangan yang sesuai dengan suhu yang diukur. Dengan tingkat akurasi yang cukup tinggi, sensor ini mampu mengukur suhu dalam rentang -55°C hingga 150°C . Selain itu, LM35 dapat dengan mudah diintegrasikan dengan mikrokontroler atau Arduino, sehingga sering digunakan dalam berbagai perangkat instrumentasi pengukuran (Imam Muammarul, 2019).

Tabel 2. 5 Spesifikasi Sensor DS18B20

Fitur	Spesifikasi
Tegangan <i>Supply</i>	4V – 30V
Arus Operasi	60mA
Rentang Pengukuran	$-55 - 155^{\circ}\text{C}$
<i>Form Factor</i>	3-Wire, Probe

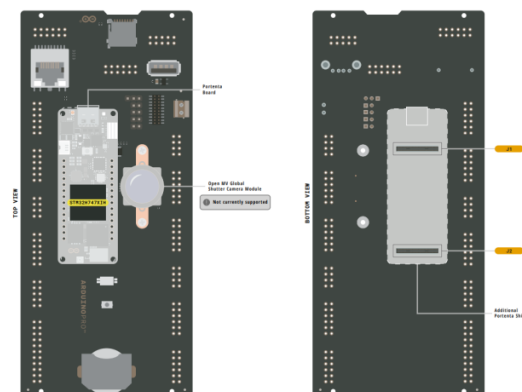
2.5.4. Sensor Ultrasonik HCSR-04



Gambar 2. 14 Sensor HCSR-04 (Arifin et al., 2022)

Sensor ultrasonik HC-SR04 yang ditampilkan pada Gambar 1 beroperasi dengan frekuensi 40Hz. Sensor ini berfungsi untuk mengukur jarak antara objek dengan sensor itu sendiri. HC-SR04 memiliki empat pin utama, yaitu Vcc, Trigger, Echo, dan Ground. Adapun spesifikasi dari sensor ultrasonik HC-SR04, yaitu bekerja pada tegangan DC 5V dengan arus operasional sebesar 15mA, serta menggunakan frekuensi kerja 40Hz. Sensor ini mampu mengukur jarak dengan rentang antara 2 cm hingga 4 meter, dengan sudut pengukuran sekitar 15 derajat. Selain itu, sensor ini menerima sinyal pemicu berupa pulsa TTL 10 mikrodetik (Arifin et al., 2022).

2.6 Portenta H7 Breakout



Gambar 2. 15 Portenta Breakout H7(Input et al., 2023)

Portenta H7 adalah bagian dari keluarga Arduino yang dirancang untuk aplikasi tingkat lanjut seperti AI, IoT, dan robotika. Portenta H7 yang ditunjukkan Gambar 2.15 dibangun dengan mikroprosesor STM32H747XI yang memiliki dua inti (Cortex-M7 dan Cortex-M4), board ini mampu menjalankan kode tingkat tinggi secara bersamaan dengan tugas real-time. Portenta H7 mendukung Wi-Fi, Bluetooth 5.1, dan akselerator GPU Chrom-ART untuk aplikasi grafis. Breakout board untuk Portenta H7 memberikan akses ke semua sinyal dari konektor berdensitas tinggi, sehingga memudahkan pengembangan prototipe hardware.

Fitur tambahan termasuk konektor JTAG untuk debugging dan soket kamera OpenMV untuk pengembangan aplikasi visi mesin. Breakout board ini ideal untuk menguji koneksi dan kapabilitas perangkat eksternal saat mengembangkan sistem berbasis Portenta H7 (Input et al., 2023).

Tabel 2. 6 Spesifikasi Portenta H7 Breakout

Komponen	Spesifikasi
Mikroprosesor	STM32H747XI, dual-core Cortex-M7 (480 MHz) + Cortex-M4 (240 MHz)
RAM	8 MB SDRAM
Memori Flash	16 MB NOR Flash
Konektivitas Nirkabel	Murata 1DX dengan Wi-Fi 802.11 b/g/n dan Bluetooth 5.1
GPU	Chrom-ART graphical accelerator dengan encoder/decoder JPEG
Port USB	USB-C (mendukung komunikasi serial, daya, dan video output via DisplayPort)
Konektor GPIO	80-pin high-density connector untuk akses ke semua sinyal board
Debugging	JTAG konektor untuk debugging perangkat keras
Sumber Daya	Input 5V via USB-C atau 3.3V/5V melalui konektor breakout
Dimensi	102 x 25 mm

2.7 Pompa Peristaltik



Gambar 2. 16 Pompa Peristaltik (Maryanto et al., 2018)

Pompa Peristaltik yang ditunjukkan Gambar 2.16 adalah jenis pompa yang bekerja berdasarkan prinsip peristaltik, yaitu pergerakan gelombang atau kontraksi yang terjadi pada saluran fleksibel untuk memindahkan cairan. Pompa ini terdiri dari sebuah tabung fleksibel yang dilalui oleh cairan, dan rol-rol yang berputar di sepanjang tabung tersebut. Ketika rol-rol ini berputar, mereka menekan tabung di beberapa titik tertentu, menciptakan ruang hampa di belakangnya yang kemudian menarik cairan untuk mengalir ke dalam tabung. Setelah rol bergerak melewati titik tersebut, tabung kembali mengembang, memaksa cairan untuk mengalir ke depan (Maryanto et al., 2018).

2.8 Relay



Gambar 2. 17 *Relay* (Nugraha, 2019)

Relay adalah perangkat elektromagnetik yang berfungsi sebagai saklar atau pemutus hubungan listrik yang dikendalikan oleh sinyal elektrik seperti yang ditunjukkan Gambar 2.17. Pada dasarnya, *relay* bekerja dengan prinsip induksi elektromagnetik, di mana adanya arus listrik yang mengalir melalui kumparan akan menghasilkan medan magnet yang menarik atau menolak lengan penghubung untuk membuka atau menutup kontak-kontak *relay*. Komponen utama dari relay terdiri dari kumparan (*coil*), kontak (*contact*), dan mekanisme penggerak (*armature*) yang dapat menghubungkan atau memutuskan jalur arus listrik. Relay dapat dibedakan menjadi beberapa jenis berdasarkan cara kerjanya, seperti SPDT (*Single Pole Double Throw*), DPDT (*Double Pole Double Throw*),

atau SSR (*Solid State Relay*), yang masing-masing memiliki karakteristik dan aplikasi yang berbeda sesuai dengan kebutuhan sistem (Nugraha, 2019).

2.9 Internet of Things (IoT)



Gambar 2. 18 Konsep IoT (Efendi, 2018)

Internet of Things (IoT) adalah konsep yang bertujuan untuk memaksimalkan manfaat dari konektivitas internet yang selalu terhubung seperti yang ditunjukkan Gambar 2.18. Secara umum, IoT merujuk pada objek-objek yang dapat diidentifikasi secara unik sebagai representasi virtual dalam suatu sistem berbasis internet. Cara kerja IoT melibatkan interaksi antar mesin yang terhubung secara otomatis, tanpa perlu campur tangan manusia, dan dapat terjadi dalam jarak yang sangat jauh. Untuk mendukung cara kerja tersebut, internet berfungsi sebagai penghubung antar mesin, sementara pengguna hanya berperan sebagai pengatur dan pemantau jalannya perangkat(, Suhardi, 2018) . Dengan IoT, berbagai tugas dapat dilakukan lebih cepat, mudah, dan efisien (Efendi, 2018). Sistem dasar IoT terdiri dari tiga komponen utama, yaitu:

- a. Perangkat keras (*Things*)
- b. Koneksi internet
- c. *Cloud Data Center*

2.10 Quality of Service (QoS)

Quality of Services (QoS) merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan suatu untuk mendefinisikan karakteristik

dan sifat dari suatu service. *Quality of Service* (QoS) digunakan untuk mengukur sekumpulan atribut performa yang telah diasosiasikan dengan suatu service, dengan parameter berupa delay, jitter, packet loss dan throughput(Rachmadi et al., 2021).

1. *Delay*

Delay adalah waktu yang dibutuhkan paket untuk mencapai tujuan, karena adanya antrian, atau mengambil rute lain untuk menghindari kemacetan(Rachmadi et al., 2021). Menurut versi *TIPHON (Telecommunication and Internet Protocol Harmonization Over Network)* standarisasi nilai delay/latency terlihat pada Tabel 2.6.

Tabel 2. 7 Parameter Delay

Kategori	Besar Delay
Sangat Bagus	<150ms
Bagus	150ms – 300ms
Sedang	300ms – 450ms
Buruk	>450ms

Menurut Tabel 2.7 bahwa standarisasi *delay* yang dikategorikan sangat bagus berada di bawah 150ms, kategori bagus berada diantara 150ms-300ms, kategori sedang berada diantara 300ms-450ms, dan kategori buruk diatas 450ms.

2. *Jitter*

Jitter didefinisikan sebagai perubahan *latency* pada suatu periode. *Jitter* merupakan penundaan perpariasi dari waktu ke waktu. *Jitter* juga didefinisikan sebagai gangguan pada komunikasi digital maupun analog yang disebabkan oleh perubahan sinyal karena referensi posisi waktu. Adanya *jitter* ini dapat mengakibatkan hilangnya data, terutama pada pengiriman data dengan kecepatan

tinggi. Secara umum terdapat empat kategori penurunan kualitas jaringan berdasarkan nilai *jitter* sesuai versi standarisasi TIPHON (*Telecommunication and Internet Protocol Harmonization Over Network*) seperti pada Tabel 2.3(Rachmadi et al., 2021).

Tabel 2. 8 Parameter Jitter

Kategori	Besar Jitter
Sangat Bagus	0
Bagus	1ms – 75ms
Sedang	76ms – 125ms
Buruk	126ms – 225ms

Menurut Tabel 2.8 bahwa standarisasi *jitter* yang dikategorikan sangat bagus adalah bernilai 0, artinya tidak ada perubahan *latency* dari setiap periode, kategori bagus berada diantara 1ms-75ms, kategori sedang berada diantara 76ms-125ms, dan kategori buruk diatas 126ms-225ms.

3. *Packet loss*

Packet loss didefinisikan sebagai kegagalan transmisi paket data mencapai tujuannya. Kegagalan paket tersebut mencapai tujuan dapat disebabkan oleh beberapa kemungkinan(Rachmadi et al., 2021).

Tabel 2. 9 Parameter *Packet Loss*

Kategori	Besar Packet Loss
Sangat Bagus	0% - 2%
Bagus	3% - 14%
Sedang	15% - 24%
Buruk	>25%

Menurut Tabel 2.9 bahwa standarisasi *packet loss* yang dikategorikan sangat bagus berada diantar 0%-2%, kategori bagus berada diantara 3%-14%,

kategori sedang berada diantara 15%-24%, dan kategori buruk diatas 25%.

4. *Throughput*

Throughput adalah kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data. *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada destinasi selama *interval* waktu tertentu dibagi oleh durasi *interval* waktu tersebut. Biasanya *throughput* selalu dikaitkan dengan *bandwidth*.

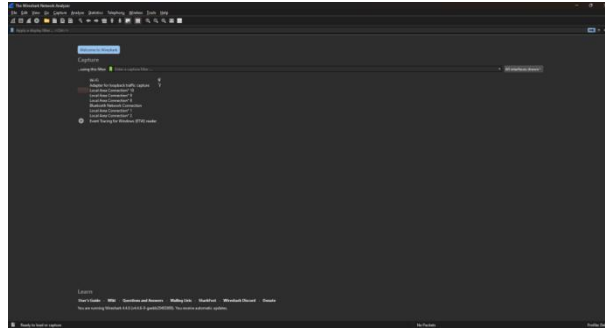
Tabel 2. 10 Parameter Throughput

Kategori	Throughput
Sangat Bagus	>2.1 Mbps
Bagus	1.2 Mbps – 2.1 Mbps
Sedang	338 Kbps – 1200 Kbps
Buruk	0 Kbps – 338 Kbps

Menurut Tabel 2.10 bahwa standarisasi *throughput* yang dikategorikan sangat bagus berada di atas 2.1Mbps, kategori bagus berada diantara 1.2Mbps-2.1Mbps, kategori sedang berada diantara 700 Kbps – 1200 Kbps, dan kategori buruk diantara 700 Kbps – 1200 Kbps.

2.11 *Wireshark*

Wireshark adalah perangkat lunak yang digunakan untuk menganalisis komunikasi data dalam jaringan. Sebagai alat *network analyzer*, *wireshark* berfungsi untuk menangkap dan memeriksa paket data yang sedang melintas di dalam sistem jaringan.



Gambar 2. 19 Tampilan Wireshark (Dokumentasi Pribadi)

2.12 Penelitian Terkait

Implementasi kendali *Artificial Neural Network* (ANN) pada Hidroponik telah banyak digunakan dalam berbagai penelitian dengan berbagai macam penelitian yang dipublikasikan pada sebuah jurnal. Dari beberapa penelitian tersebut penulis jadikan sebagai perbandingan pada tabel 2.11 sehingga dapat memperkuat pemahaman penulis dalam mengidentifikasi dan menganalisis kendali ANN pada penelitian ini.

Tabel 2. 11 Jurnal Penelitian Terkait

No	Judul Jurnal	Nama Peneliti	Tempat dan Tahun Penelitian	Pembahasan jurnal
1.	Rancang Bangun Sistem Kendali Zat Terlarut dalam Air Hidroponik Berbasis	Syakira Andriyani, Tri Susijanto, Dimas Rasmana, Dimas Adiputra	Laboratorium Teknik Elektro Telkom University Surabaya, 2024	Penelitian ini berhasil mengintegrasikan sensor TDS dengan Arduino untuk mengukur kualitas air meski terdapat kendala akibat usia sensor. Estimasi nutrisi menggunakan JST dengan MATLAB menghasilkan akurasi training 0,93, yang kemudian diterapkan pada Arduino untuk mengontrol durasi pompa. Dari 30 data pengujian, sistem mencapai akurasi 86,9% dengan nilai error

No	Judul Jurnal	Nama Peneliti	Tempat dan Tahun Penelitian	Pembahasan jurnal
				13,1%, MSE sebesar 87,6, dan standar deviasi 12,4.
2.	Implementasi Kontrol Nutrisi dan pH Pada Hidroponik Cerdas Berbasis Arduino dan JST	M Naufal Zul Hazmi, Raden Sumiharto	Laboratorium Teknik Elektronika Instrumentasi UGM, 2023	Penelitian ini mengembangkan sistem kontrol otomatis untuk nutrisi dan pH pada hidroponik NFT berbasis JST. JST memprediksi output aktuator pompa (pH Up, pH Down, TDS) berdasarkan input sensor pH dan TDS. Penelitian lanjutan menggunakan model JST untuk mengontrol otomatis nutrisi dan pH pada hidroponik NFT, dengan memprediksi output aktuator pompa untuk pH Up, pH Down, dan nutrisi A dan nutrisi B berdasarkan input sensor pH dan TDS.
3.	Implementasi Alat Pemberi Nutrisi pada Hidroponik berbasis Arduino	Yoga Prayoga	Laboratorium Informatika Universitas Darmajaya, 2019	Penelitian ini lebih fokus pada pengendalian otomatis nutrisi dengan menggunakan mikrokontroler ATmega 2560, sensor TDS, dan sensor water level untuk memantau kualitas nutrisi dan ketinggian air. Sistem ini mengaktifkan buzzer sebagai peringatan jika nilai TDS tidak sesuai dan mengendalikan relay berdasarkan ketinggian air.
4.	Pengembangan Sistem Kontrol Nutrisi	Renny Eka Putri, Hammam Mananda	Laboratorium Keteknikan Pangan	Penelitian terdahulu mengembangkan sistem hidroponik rakit apung

No	Judul Jurnal	Nama Peneliti	Tempat dan Tahun Penelitian	Pembahasan jurnal
	Budidaya Hidroponik Berbasis IoT (Internet of Things) Sawi Samhong (Brassicasinensis L.)	Harahap, dan Iriwad Putri	Universitas Andalas 2023	berbasis IoT dengan sensor TDS untuk memantau dan mengontrol penambahan nutrisi AB mix secara otomatis melalui aplikasi Blynk.
5.	Sistem Pemantauan dan Pengendalian Nutrisi, Suhu, dan Tinggi Air pada Pertanian Hidroponik Berbasis Website	Yuga Hadfridar Putra, Dedi Triyanto, Suhardi	Laboratorium Sistem Komputer UNTAN 2018	Penelitian ini mengembangkan sistem otomatis berbasis NodeMCU ESP8266 untuk memantau dan mengendalikan nutrisi, suhu, dan tinggi air hidroponik. Menggunakan sensor DS18B20, HC-SR04, dan larutan nutrisi, sistem bekerja nirkabel untuk mengaktifkan pompa, kipas, atau heater sesuai batas yang ditentukan. Hasilnya, efisiensi pengelolaan hidroponik meningkat.

Sumber: Hasil Studi Literatur, 2024

Penelitian sebelumnya yang dilakukan oleh Syakira Andriyani dkk, penelitian ini melakukan kontrol dengan metode ANN namun hanya mengintegrasikan sistem pengukuran dan kontrol nutrisi PPM AB Mix saja tanpa melakukan kontrol terhadap pH. Penelitian yang akan dilakukan yaitu dengan menambahkan parameter pH untuk pemantauan dan kontrol otomatis. Selain itu, analisis error yang akan dilakukan meliputi MAPE untuk prediksi kinerja pompa dan analisis kestabilan model menggunakan standar deviasi. Sistem yang dikembangkan juga terhubung secara real-time ke IoT.

Pada penelitian sebelumnya oleh M Naufal Zul Hasmi dkk, sistem kontrol

otomatis untuk hidroponik NFT berbasis ANN memprediksi keluaran berupa kondisi on/off pompa untuk pH Up, pH Down, dan nutrisi berdasarkan input sensor pH dan TDS. Pendekatan ini hanya mengaktifkan atau menonaktifkan pompa tanpa memperhitungkan durasi waktu nyala pompa. Sebagai pengembangan lebih lanjut, penelitian ini akan membuat sistem yang memprediksi durasi nyala pompa dalam satuan detik sebagai output.

Penelitian yang dilakukan oleh Yoga Prayoga dkk, lebih fokus pada pengendalian otomatis nutrisi menggunakan mikrokontroler ATmega 2560, sensor TDS, dan sensor water level untuk memantau kualitas nutrisi dan ketinggian air. Penelitian yang akan dilakukan menggunakan metode Artificial Neural Network (ANN) untuk pengontrolan pH dan PPM nutrisi AB Mix secara otomatis. Dengan ANN, sistem dapat memprediksi durasi nyala pompa nutrisi dalam satuan detik berdasarkan data sensor PPM dan pH.

Penelitian terdahulu oleh Renny Eka Putri dkk, mengembangkan sistem hidroponik rakit apung berbasis IoT dengan sensor TDS untuk memantau dan mengontrol penambahan nutrisi AB mix secara otomatis melalui aplikasi Blynk. Penelitian yang akan dilakukan menggunakan sistem hidroponik NFT (*Nutrient Film Technique*) yang tidak hanya mengontrol penambahan nutrisi AB mix tetapi juga mengontrol parameter pH secara otomatis. Metode pengontrolan yang digunakan adalah *Artificial Neural Network* (ANN), yang mampu memprediksi durasi nyala pompa untuk penyesuaian nutrisi dan pH berdasarkan input dari sensor TDS dan pH. Data yang diperoleh akan dikirimkan secara IoT menggunakan *NodeRed* melalui protokol MQTT.

Penelitian terdahulu oleh Yuga Hadfridar Putra dkk, mengembangkan

sistem otomatis berbasis *NodeMCU* ESP8266 untuk memantau dan mengendalikan nutrisi dan suhu hidroponik. Sistem menggunakan sensor DS18B20 dan sensor larutan nutrisi, serta bekerja secara nirkabel untuk mengaktifkan pompa, kipas, atau heater sesuai dengan batas yang telah ditentukan. Sebagai pengembangan lebih lanjut, penelitian ini akan fokus pada pengontrolan nutrisi (ppm AB Mix) dan pH menggunakan metode *Artificial Neural Network* (ANN). ANN akan digunakan untuk memprediksi durasi kerja pompa secara presisi berdasarkan data sensor TDS dan pH. Data pH dan PPM AB Mix akan dikirimkan secara IoT menggunakan Portenta H7.