

**PERAMALAN KINERJA GENSET DI PONDOK PESANTREN  
HARAKATUL QUR'AN MENGGUNAKAN METODE  
CONVOLUTIONAL NEURAL NETWORK**

**DRAFT TUGAS AKHIR**

Karya Ilmiah sebagai salah satu syarat untuk menyelesaikan jenjang strata satu  
(S-1) di Departemen Teknik Elektro, Fakultas Teknik, Universitas Andalas

Oleh

Fadly Ihsan Andanny  
1810953001

Pembimbing (Utama)  
Zaini, Ph.D  
NIP. 197603212001121003



**Program Studi Sarjana  
Departemen Teknik Elektro  
Fakultas Teknik  
Universitas Andalas  
2025**

# LEMBAR PENGESAHAN

Tugas Akhir

Peramalan Kinerja Genset di Pondok Pesantren Harkatul Qur'an menggunakan  
Metode Convolutional Neural Network

Oleh

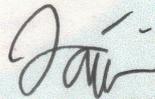
Fadly Ihsan Andanny

1810953001

Departemen Teknik Elektro Fakultas Teknik  
Universitas Andalas

Disetujui pada Tanggal : 30 - 07 - 2025

Pembimbing (Utama)



Zaini, Ph.D

NIP. 197603212001121003

Mengetahui,

Ketua Departemen Teknik Elektro



  
Prof. Syafii, Ph.D

NIP. 19740505198021001

## HALAMAN PENGHARGAAN

"Bekerjalah seolah-olah kamu hidup selamanya, dan beribadahlah seolah-olah kamu mati besok." — Ali bin Abi Thalib

"*Success is not final, failure is not fatal: it is the courage to continue that counts.*" — Winston Churchill

Puji syukur penulis panjatkan ke hadirat Allah SWT, karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir ini dengan baik. Proses penyusunan tugas akhir ini tentunya tidak terlepas dari peran dan dukungan berbagai pihak yang telah memberikan bantuan, bimbingan, serta motivasi, baik secara moral maupun material.

Sebagai bentuk penghargaan dan rasa terima kasih yang tulus, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bundaku tersayang Almh. Mirnayati, A.Md.Ak, yang telah melahirkan penulis untuk melihat dunia yang ini. Maafkan penulis tidak bisa menyelesaikan tugas akhir ini sebelum ditinggal pergi oleh Bunda. Terima kasih selama hidup bunda selalu percaya ke pada penulis. *May God bring us together again someday, Amen.*
2. Ayahanda Darmawadi, yang selalu *men-support* penulis dari penulis lahir sampai sekarang, dan juga selalu memberikan masukkan di saat penulis kebingungan arah. Penulis hanya ingin mengucapkan maaf, karena tidak bisa menjadi seperti yang Ayah harapkan. *You are the greatest dad in the world.*
3. Adikku tersayang Jihan Aulia Andhina, kadang selalu mengajak penulis berantam di saat kami masih kecil, tetapi selalu perhatian dengan penulis, dan selalu menyemangati penulis dalam penyusunan tugas akhir ini. *Don't give up, you're nearly done!*
4. *Cayankcu* tersayang Mira Syuriyani, S.Si, yang selalu merepotkan penulis, tetapi dibalik itu *cayankcu* selalu mengingatkan dan *men-support* penulis dalam penyusunan tugas akhir ini. *You are the best thing that's ever been mine – Mine(Taylor's Version).*
5. Ante Yanti Lora, A.Md.Ak, yang telah hadir sebagai sosok bunda dalam hidup penulis. Terima kasih telah menerima penulis dengan penuh kasih sayang dan kesabaran, serta senantiasa memberikan dukungan tanpa pamrih selama ini. Doa dan dorongan Ante adalah bagian penting dalam perjalanan ini. *You're not my mother by blood, but I love you as if you were.*

6. Saudara-saudara penulis Muhammad Aqra Selmi Rinaldy, Achmad Zikrie Selmi Rinaldy, dan Aqela Jazilatur Rahma Rinaldy. Yang walau tidak terikat darah, tetapi telah menjadi bagian dari hidup penulis sebagai keluarga sejati. Terima kasih atas kebersamaan, canda, semangat, dan dukungan yang kalian berikan selama ini. *Step-sibling is just a label, because in my heart, you are my true siblings.*
7. Sepuh segala hal Yesta Rasyid, S.T. kawan dekat penulis, yang selalu tukar pikiran tentang segala hal, apalagi untuk tugas akhir ini. Walaupun sering mengajak penulis main *game*, sepuh satu ini selalu memberikan saran untuk tugas akhir penulis. *R stands for red flag, obviously.*
8. Fahrhanul Aderi, S.T, kawan satu bimbingan yang selalu menolong penulis dalam penyusunan tugas akhir ini, dan selalu mentransferkan ilmu-ilmunya kepada penulis. *You can seek knowledge, but finding a humble, intelligent friend is a rare fortune.*
9. Uda Cresti, Uda Reka, Uda Anjang, Uda Jepang, Uda Aan, Uda Donny, Uni Gewe dan Uda-Uni KATUA (Keluarga Alumni Teknik Universitas Andalas) lainnya yang tidak dapat penulis sebutkan satu per satu. Selalu mengingatkan penulis untuk menyelesaikan tugas akhir dan selalu mengasih ilmu-ilmunya baik akademik maupun non akademik.
10. Ucapan terima kasih juga penulis sampaikan kepada seluruh keluarga besar yang tidak dapat disebutkan satu per satu. Terima kasih atas segala bentuk doa, dukungan, perhatian, dan semangat yang telah diberikan selama ini. Kehadiran dan kebaikan kalian, dalam berbagai cara, telah menjadi bagian penting dalam perjalanan penulis hingga titik ini.
11. Terima kasih yang sebesar-besarnya penulis sampaikan kepada seluruh teman-teman yang telah menjadi bagian penting dalam perjalanan ini. Terima kasih atas kebersamaan, tawa, diskusi panjang, saling bantu di saat sulit, dan semangat yang tak pernah padam. Kalian bukan hanya teman kuliah, tapi juga saudara dalam perjuangan. Semoga persahabatan ini tetap terjaga meski langkah kita akan membawa ke arah yang berbeda.

Padang, Juli 2025

Penulis

## LEMBAT PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini :

Nama : Fadly Ihsan Andanny  
NIM : 1810953001  
Program Studi : S1 Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Perguruan Tinggi : Universitas Andalas

Dengan ini saya menyatakan bahwa tugas akhir berjudul “Peramalan Kinerja Genset di Pondok Pesantren Harakatul Qur’an menggunakan Metode *Convolutional Neural Network*” merupakan hasil kerja saya sendiri dan bebas dari unsur plagiarisme. Pernyataan ini saya buat dengan sebenar-benarnya, dan saya siap menerima sanksi dari Departemen Teknik Elektro, Fakultas Teknik, Universitas Andalas jika di kemudian hari ditemukan pelanggaran atau ketidaksesuaian terhadap isi pernyataan ini.

Oleh karena itu, pernyataan keaslian ini saya buat secara sadar dan tulus, tanpa adanya tekanan atau paksaan dari pihak mana pun.

Padang, 16 Juli 2025

Penulis



Fadly Ihsan Andanny

## RIWAYAT HIDUP

Fadly Ihsan Andanny lahir di Duri, Bengkalis pada tanggal 17 Mei 2000, sebagai anak sulung dari tiga bersaudara, putra dari Bapak Darmawardi dan almarhumah Ibu Mirnayati. Pendidikan dasarnya diselesaikan di SDIT Jami'atul Muslimin pada tahun 2012, kemudian melanjutkan ke SMP Negeri 2 Dumai dan lulus pada tahun 2015. Setelah itu, ia menyelesaikan pendidikan menengah atas di SMA Negeri 1 Dumai pada tahun 2018 dan di tahun yang sama diterima di Program Studi S1 Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik, Universitas Andalas.

Selama menempuh pendidikan di perguruan tinggi, penulis aktif dalam berbagai organisasi, di antaranya Badan Eksekutif Mahasiswa Keluarga Mahasiswa Universitas Andalas (BEM KM UNAND) sebagai Staf Audit Internal pada periode 2019–2020, serta menjadi Koordinator Operasional di organisasi NEOTELEMETRI pada tahun 2021. Selain itu, penulis juga mengikuti program Magang Bersertifikat sebagai bagian dari Merdeka Belajar Kampus Merdeka (MBKM) di PT Semen Padang selama enam bulan, dari Januari hingga Agustus 2022.



Judul	Peramalan Kinerja Genset di Pondok Pesantren Harkatul Qur'an Menggunakan Metode <i>Convolutional Neural Network</i>	Fadly Ihsan Andanny
Program Studi	Teknik Elektro	1810953001
Fakultas Teknik Universitas Andalas		
<p>Abstrak</p> <p>Ketergantungan pada energi listrik untuk aktivitas manusia tidak dapat disangkal, dengan konsumsi energi listrik di Indonesia meningkat rata-rata sebesar 6,96% antara tahun 2013 dan 2021. Sistem tenaga darurat berbasis generator diesel digunakan untuk mengambil alih beban-beban kritis selama pemadaman listrik, meskipun keandalannya dan konfigurasi sangat penting dalam pemadaman yang lebih lama. Pembangkit listrik berbahan bakar fosil harus beroperasi pada rentang beban sedang dan puncak untuk menstabilkan jaringan listrik, yang menyebabkan tekanan <i>termomekanis</i> pada komponen inti seperti generator. Analisis getaran dan suhu merupakan teknik pemantauan penting untuk mesin yang berputar, memungkinkan pemeliharaan keandalan dan kinerja generator untuk menghindari pemadaman yang tidak terduga dan mengurangi konsumsi bahan bakar. Alat pengukur seperti sensor daya, getaran, dan suhu sangat penting untuk memantau kinerja mesin. Teknologi IoT memfasilitasi koneksi mesin ke internet, memungkinkan pengumpulan data dari berbagai sensor dan perangkat pemantauan secara <i>real-time</i>. Data yang dikumpulkan dapat dianalisis dengan algoritma pembelajaran mesin untuk memperoleh informasi berguna tentang kinerja mesin. Peramalan jangka pendek berdasarkan data masa lalu dan masa kini dapat memprediksi pemeliharaan generator, dengan teknologi CNN yang mengidentifikasi data dari sensor yang terpasang pada generator untuk menunjukkan tanda-tanda awal kerusakan peralatan.</p> <p>Kata Kunci : Kinerja, Generator, Peramalan, CNN</p>		

<i>Title</i>	<i>Performance Forecasting of Gensets at Pondok Pesantren Harakatul Qur'an using Convolutional Neural Network Method.</i>	Fadly Ihsan Andanny
<i>Mayor</i>	<i>Electrical Engineering Department</i>	1810953001
<i>Engineering Faculty Andalas University</i>		
<i>Abstract</i>		
<p><i>The reliance on electrical energy for human activities is undeniable, with Indonesia's electrical energy consumption increasing by 6.96% on average between 2013 and 2021. Diesel generator-based emergency power systems are deployed to take over the critical loads during power outages, albeit their reliability and configuration are imperative in longer outages. Fossil fuel power plants must operate at medium and peak load ranges to stabilize the electricity grid, leading to thermomechanical stress on core components such as generators. Vibration and temperature analysis are essential monitoring techniques for rotating machines, enabling the maintenance of generator reliability and performance to avoid unexpected outages and reduce fuel consumption. Measuring devices such as power, vibration, and temperature sensors are vital for monitoring engine performance. IoT technology facilitates machines' connection to the internet, enabling data collection from various sensors and monitoring devices in real-time. The collected data can be analyzed with machine learning algorithms to obtain useful information about machine performance. Short-term forecasting based on past and present data can predict generator maintenance, with CNN technology identifying data from sensors installed on the generator to show early signs of damage to the equipment.</i></p> <p><i>Key Word : Performance, Generators, Forecasting, CNN</i></p>		

ABSTRACT

KEDJAJAAN

UNTUK

BANGSA

## DAFTAR ISI

<b>LEMBAR PENGESAHAN .....</b>	<b>II</b>
<b>HALAMAN PENGHARGAAN .....</b>	<b>III</b>
<b>LEMBAT PERNYATAAN KEASLIAN .....</b>	<b>V</b>
<b>RIWAYAT HIDUP .....</b>	<b>VI</b>
<b>ABSTRAK .....</b>	<b>VII</b>
<b>ABSTRACT .....</b>	<b>VIII</b>
<b>DAFTAR ISI.....</b>	<b>IX</b>
<b>PRAKATA .....</b>	<b>XII</b>
<b>DAFTAR TABEL .....</b>	<b>XIII</b>
<b>DAFTAR GAMBAR.....</b>	<b>XIV</b>
<b>DAFTAR SIMBOL .....</b>	<b>XVI</b>
<b>DAFTAR LAMPIRAN.....</b>	<b>XVII</b>
<b>DAFTAR ISTILAH .....</b>	<b>XVIII</b>
<b>DAFTAR SINGKATAN.....</b>	<b>XXII</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. <b>LATAR BELAKANG .....</b>	<b>1</b>
1.2. <b>RUMUSAN MASALAH .....</b>	<b>2</b>
1.3. <b>TUJUAN PENELITIAN .....</b>	<b>3</b>
1.4. <b>MANFAAT PENELITIAN .....</b>	<b>3</b>
1.5. <b>BATASAN MASALAH .....</b>	<b>3</b>
1.6. <b>SISTEM PENULISAN .....</b>	<b>3</b>
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>5</b>
2.1. <b><i>DIESEL GENERATOR SET</i> .....</b>	<b>5</b>
2.2. <b>GETARAN .....</b>	<b>10</b>
2.3. <b>SENSOR .....</b>	<b>11</b>
2.3.1. <b>Sensor Tegangan (ZMPT101B).....</b>	<b>12</b>
2.3.2. <b>Sensor Arus (ACS712).....</b>	<b>14</b>
2.3.3. <b>Sensor Getaran dan Suhu (MPU6050).....</b>	<b>15</b>
2.4. <b>MIKROKONTROLER .....</b>	<b>17</b>
2.4.1. <b>ESP32.....</b>	<b>18</b>
2.4.2. <b>Arduino IDE.....</b>	<b>20</b>

2.5.	<i>INTERNET OF THINGS (IoT)</i> .....	20
2.5.1.	<i>Message Queuing Telemetry Transport (MQTT)</i> .....	23
2.5.2.	Node-RED .....	24
2.5.3.	<i>Database</i> .....	24
2.5.4.	MySQL .....	24
2.5.5.	Phyton .....	25
2.6.	<i>FORECASTING</i> .....	25
2.6.1.	<i>Convolutional Neural Network (CNN)</i> .....	26
2.7.1.1.	<i>Convolution Layer</i> .....	27
2.7.1.2.	<i>Pooling Layer</i> .....	27
2.7.1.3.	<i>Fully Connected Layer</i> .....	28
2.7.1.4.	<i>Rectified Linear Units (ReLU)</i> .....	28
2.6.2.	<i>Perhitungan Akurasi Forecasting</i> .....	29
2.7.2.1.	<i>Mean Square Error (MSE)</i> : .....	29
2.7.2.2.	<i>Mean Absolute Percentage Error (MAPE)</i> : .....	29
2.7.2.3.	<i>Mean Percentage Error (MPE)</i> : .....	29
<b>BAB III METODE PENELITIAN</b> .....		<b>31</b>
3.1.	JENIS PENELITIAN .....	31
3.2.	TEMPAT DAN WAKTU PELAKSANAAN .....	31
3.3.	TAHAP PELAKSANAAN PENELITIAN .....	32
3.4.	PERANCANGAN SISTEM .....	34
3.5.	PERANCANGAN <i>MONITORING</i> DENGAN ESP32 .....	37
3.6.	PERANCANGAN MQTT .....	39
3.7.	PERAMALAN TINGKAT KERUSAKAN .....	39
<b>BAB IV HASIL DAN ANALISA</b> .....		<b>41</b>
4.1.	IMPLEMENTASI DESAIN SISTEM <i>MONITORING</i> .....	41
4.2.	PROSES TRANSMISI DATA HASIL PEMANTAUAN GENSET SECARA <i>REAL-TIME</i> .....	44
4.3.	TAHAPAN PENYIMPANAN DATA <i>MONITORING</i> KE <i>DATABASE</i> MYSQL DI <i>LOCALHOST</i> .....	45
4.4.	IMPLEMENTASI <i>DATABASE</i> UNTUK PERAMALAN MENGGUNAKAN METODE CNN .....	46
4.5.	ANALISIS HASIL PERAMALAN JANGKA PENDEK <i>DAYA MONITORING</i> GENSET MENGGUNAKAN METODE CNN .....	50
4.6.	ANALISIS HASIL PERAMALAN JANGKA PENDEK <i>SUHU MONITORING</i> GENSET MENGGUNAKAN METODE CNN .....	54
4.7.	ANALISIS HASIL PERAMALAN JANGKA PENDEK <i>GETARAN MONITORING</i> GENSET MENGGUNAKAN METODE CNN .....	58
<b>BAB V SIMPULAN DAN SARAN</b> .....		<b>62</b>
5.1.	SIMPULAN .....	62

5.2. SARAN..... 62

**DAFTAR PUSTAKA..... 63**

**LAMPIRAN 1 PROGRAM MONITORING (C++/ARDUINO)..... 67**

**LAMPIRAN 2 PROGRAM FORECASTING (PYTHON/GOOGLE COLAB)**  
**..... 74**



## PRAKATA

Puji syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan tugas akhir ini dengan baik. Penyusunan tugas akhir ini tentunya tidak terlepas dari bantuan, dukungan, dan kontribusi berbagai pihak, baik dalam bentuk moral maupun material. Oleh karena itu, dengan segala hormat dan rasa terima kasih yang tulus, penulis menyampaikan penghargaan yang sebesar-besarnya kepada:

1. Bapak Prof. Syafii, S.T., M.T., Ph.D., selaku Ketua Departemen Teknik Elektro, Fakultas Teknik, Universitas Andalas, atas dukungan dan fasilitas yang diberikan selama penulis menempuh studi.
2. Bapak Ir. Novizon, M.Eng., Ph.D., selaku Sekretaris Departemen Teknik Elektro, Fakultas Teknik, Universitas Andalas, atas bantuan dan arahnya selama proses penyusunan tugas akhir ini.
3. Bapak Primas Emeraldi, S.T., M.T., selaku Ketua Program Studi S1 Teknik Elektro, Fakultas Teknik, Universitas Andalas, atas dukungan dan motivasi yang diberikan kepada penulis.
4. Bapak Zaini, Ph.D., selaku dosen pembimbing, yang telah memberikan bimbingan, arahan, motivasi, serta ilmu yang sangat berharga selama penyusunan tugas akhir ini.
5. Bapak Dr. Darwison, S.T., M.T. dan Bapak Prof. Syafii, S.T., M.T., Ph.D., selaku dosen penguji, atas saran, kritik, dan masukan yang membangun dalam rangka penyempurnaan tugas akhir ini.
6. Seluruh dosen dan staf Departemen Teknik Elektro, Fakultas Teknik, Universitas Andalas, yang telah memberikan ilmu, bantuan, serta pelayanan yang sangat membantu selama masa perkuliahan hingga penyusunan tugas akhir ini.

Padang, Juli 2025

Penulis

## DAFTAR TABEL

<b>Tabel 2. 1</b> Kriteria Zona Evaluasi Kelayakan Getaran.....	11
<b>Tabel 3. 1</b> Spesifikasi ESP32 DEVKIT V1 .....	19
<b>Tabel 3. 2</b> Spesifikasi ZMPT101B .....	13
<b>Tabel 3. 3</b> Spesifikasi ACS712.....	15
<b>Tabel 3. 4</b> Spesifikasi MPU6050.....	16
<b>Tabel 4. 1</b> Hasil Peramalan Monitoring Genset .....	49
<b>Tabel 4. 2</b> Tabel Komparasi Daya Prediksi dan Daya Aktual Genset .....	51
<b>Tabel 4. 3</b> Tabel Hasil Perbandingan Suhu Aktual dan Suhu Hasil Prediksi Genset .....	55
<b>Tabel 4. 3</b> Tabel Perbandingan Getaran Aktual dan Prediksi Genset .....	59



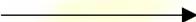
## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Diagram Konversi pada Diesel Generator Set.....	6
<b>Gambar 2. 2</b> Komponen - Komponen Genset .....	6
<b>Gambar 2. 3</b> Distribusi Probabilitas Acak dari Kekuatan dan Tekanan Stres.....	8
<b>Gambar 2. 4</b> Pegas Vertikal.....	10
<b>Gambar 2. 5</b> Blok Fungsional Sensor/ <i>Tranduser</i> .....	12
<b>Gambar 2. 6</b> Alur Proses <i>Mikrokontroler</i> .....	17
<b>Gambar 2. 7</b> Arsitektur Umum IoT .....	21
<b>Gambar 2. 8</b> Gambaran umum <i>Smart City</i> IoT Arsitektur .....	22
<b>Gambar 2. 9</b> Pub/Sub MQTT .....	23
<b>Gambar 2. 10</b> Arsitektur LeNet-5 (CNN oleh LeCun).....	26
<b>Gambar 2. 11</b> <i>Convolution Layer</i> .....	27
<b>Gambar 2. 12</b> <i>Max Pooling Layer</i> .....	28
<b>Gambar 2. 13</b> <i>Fully Connected Layer</i> .....	28
<b>Gambar 3. 1</b> Tahap Pelaksanaan Penelitian.....	32
<b>Gambar 3. 2</b> ESP32 DEVKIT V1.....	19
<b>Gambar 3. 3</b> ZMPT101B .....	13
<b>Gambar 3. 4</b> Grafik Respons ZMPT101B.....	13
<b>Gambar 3. 5</b> ACS712 .....	14
<b>Gambar 3. 6</b> Grafik Respons ACS712.....	15
<b>Gambar 3. 7</b> MPU6050.....	16
<b>Gambar 3. 8</b> Grafik Respons MPU6050.....	16
<b>Gambar 3. 9</b> Perancangan Sistem .....	35
<b>Gambar 3. 10</b> Rangkaian Alat .....	35
<b>Gambar 3. 11</b> Diagram Alir Sistem .....	36
<b>Gambar 3. 12</b> Diagram Alir Program ESP32 .....	37
<b>Gambar 3. 13</b> Diagram <i>Publish</i> dan <i>Subscribe</i> .....	39
<b>Gambar 3. 14</b> Status Mosquitto MQTT Broker.....	39
<b>Gambar 3. 15</b> Diagram Proses Peramalan dari <i>Database</i> .....	39
<b>Gambar 3. 16</b> Alur Proses Peramalan Metode CNN .....	40
<b>Gambar 4. 1</b> Diagram Hasil Perancangan Sistem <i>Monitoring</i> .....	41
<b>Gambar 4. 2</b> Rangkaian sistem <i>monitoring</i> : (a) rangkaian terdiri dari ZMPT101B, ACS712, ESP32, dan catu daya 5V; (b) MPU6050 yang diposisikan pada bagian mesin genset. ....	42
<b>Gambar 4. 3</b> Antarmuka <i>Dashboard</i> NodeRED untuk Menampilkan Sistem <i>Monitoring</i> .....	43
<b>Gambar 4. 4</b> Pengukuran Secara Manual: (a) Arus dengan <i>Clamp Meter</i> dan (b) Tegangan dengan Voltmeter .....	43
<b>Gambar 4. 5</b> Diagram <i>Publish</i> dan <i>Subscribe</i> MQTT .....	44

<b>Gambar 4. 6</b> Status Mosquitto MQTT Broker.....	44
<b>Gambar 4. 7</b> Tampilan <i>Flow Node</i> untuk Sistem <i>Monitoring</i> .....	45
<b>Gambar 4. 8</b> Tampilan phpMyAdmin di Localhost .....	46
<b>Gambar 4. 9</b> Antarmuka Proses Ekspor <i>Database</i> MySQL melalui PhpMyAdmin Localhost.....	46
<b>Gambar 4. 10</b> Cuplikan <i>File</i> CSV setelah Diekspor dari MySQL .....	47
<b>Gambar 4. 11</b> Tahapan Proses Peramalan Menggunakan Metode CNN.....	47
<b>Gambar 4. 12</b> Diagram Blok Proses Persiapan Data .....	48
<b>Gambar 4. 13</b> Diagram proses kerja metode peramalan CNN .....	48
<b>Gambar 4. 14</b> Diagram Alir Proses Pembuatan Prediksi.....	49
<b>Gambar 4. 15</b> Grafik Perbandingan Daya Aktual Genset dan Hasil Peramalan Menggunakan Metode CNN .....	50
<b>Gambar 4. 16</b> Grafik Perbandingan Suhu Genset Aktual dengan Hasil Prediksi Menggunakan Metode CNN .....	54
<b>Gambar 4. 17</b> Grafik Perbandingan Getaran Genset Aktual dan Prediksi Menggunakan Metode CNN .....	58



## DAFTAR SIMBOL

No.	Simbol	Nama	Arti
1.		Terminal	Simbol yang menunjukkan dimulainya atau berakhirnya suatu alur kerja
2.		<i>Decision</i>	Simbol yang menunjukkan proses pemilihan jalur dalam alur kerja
3.		Proses	Simbol yang menandakan pelaksanaan suatu tindakan
4.		<i>Connector</i>	Simbol penghubung yang menandai kelanjutan alur proses di bagian lain dari diagram alir pada halaman yang sama.
5.		Arus Data	Simbol yang berfungsi sebagai penunjuk aliran data atau aktivitas dalam suatu sistem.

## DAFTAR LAMPIRAN

Lampiran 1 Program Monitoring (C++/Arduino).....	67
Lampiran 2 Program Forecasting (Python/Google Colab).....	74



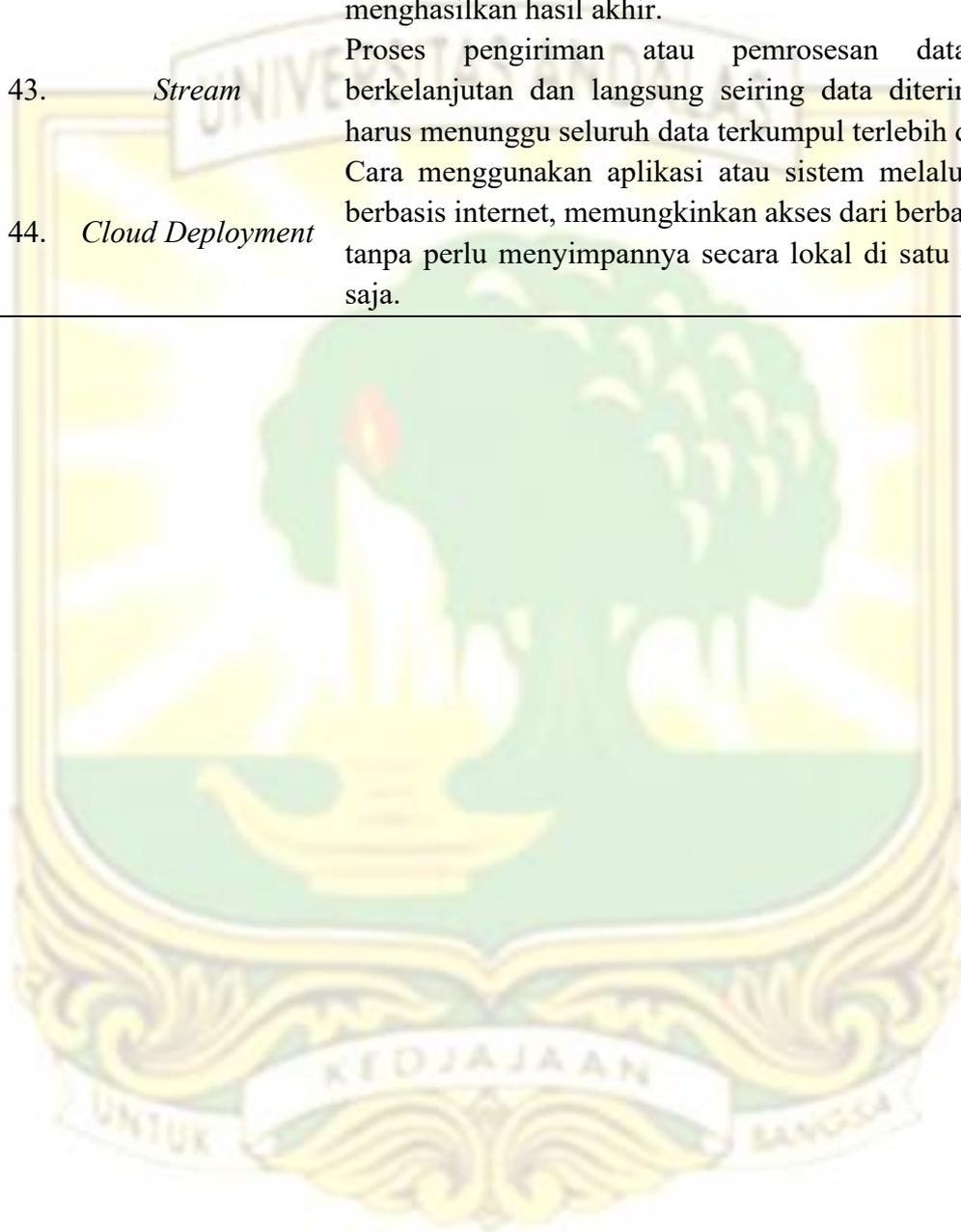
## DAFTAR ISTILAH

No.	Istilah	Pengertian
1.	<i>Delay</i>	Situasi ketika suatu proses atau peristiwa mengalami deviasi dari jadwal yang telah ditetapkan, mengakibatkan penundaan dalam respons maupun pencapaian hasil.
2.	Localhost	Alamat khusus yang digunakan oleh komputer untuk mengakses layanan atau aplikasi yang berjalan secara lokal di dalam sistemnya sendiri.
3.	<i>Excitation</i>	Merupakan proses memasok energi ke suatu bagian mesin agar dapat mulai beroperasi atau menghasilkan keluaran tertentu.
4.	<i>Konverter</i>	Alat yang berfungsi untuk mengubah suatu bentuk energi menjadi bentuk energi lain agar dapat dimanfaatkan sesuai dengan kebutuhan.
5.	<i>Monitoring</i>	Merupakan aktivitas pemantauan secara berkelanjutan untuk mengetahui kondisi, perkembangan, atau perubahan yang terjadi pada suatu objek atau sistem.
6.	<i>Automation</i>	Merupakan proses pelaksanaan tugas secara otomatis tanpa keterlibatan langsung dari manusia.
7.	<i>Aerospace</i>	Cabang ilmu dan teknologi yang fokus pada studi dan pengembangan segala hal yang berhubungan dengan penerbangan di atmosfer maupun luar angkasa.
8.	<i>Transduser</i>	Alat yang berfungsi mengonversi suatu bentuk energi ke bentuk energi lain agar dapat dideteksi, dibaca, atau diproses oleh perangkat lain.
9.	<i>Proximity</i>	Jarak dekat antara suatu benda dengan benda lainnya, dan biasanya digunakan untuk mendeteksi apakah ada benda yang mendekat.
10.	<i>Encoder</i>	Alat yang berfungsi mengubah gerakan atau posisi suatu objek menjadi sinyal yang dapat dikenali dan diproses oleh sistem atau mesin.
11.	<i>Hall Effect</i>	Merupakan fenomena saat arus listrik dan medan magnet saling berinteraksi, sehingga menimbulkan tegangan tambahan yang dapat digunakan untuk mendeteksi atau mengukur besar medan magnet maupun arus listrik.
12.	<i>Timer</i>	Perangkat atau fungsi yang digunakan untuk mengukur waktu serta mengatur kapan suatu proses harus dimulai atau dihentikan sesuai dengan pengaturan yang ditentukan.

13.	<i>Port</i>	Merupakan jalur atau media yang digunakan oleh perangkat atau aplikasi untuk melakukan pengiriman dan penerimaan data.
14.	<i>Input</i>	Segala bentuk data, energi, atau bahan yang dimasukkan ke dalam suatu sistem.
15.	<i>Output</i>	Merupakan keluaran yang dihasilkan oleh suatu sistem.
16.	<i>On-Chip Flash</i>	Tempat penyimpanan data yang tertanam, yang berfungsi untuk menyimpan program agar dapat dijalankan oleh perangkat elektronik.
17.	<i>File</i>	Sekumpulan data atau informasi yang tersimpan di dalam komputer dan dapat diakses, dibaca, maupun dimodifikasi sesuai kebutuhan pengguna.
18.	<i>Online</i>	Kondisi di mana suatu perangkat atau aplikasi sedang tersambung ke jaringan, memungkinkan terjadinya komunikasi dan pertukaran data dengan sistem atau perangkat lain.
19.	<i>Real-Time</i>	Suatu kondisi di mana sistem merespons atau menampilkan data secara langsung seiring dengan terjadinya peristiwa, tanpa penundaan yang signifikan.
20.	<i>Machine Learning</i>	Sebuah metode yang memungkinkan komputer belajar dari data dan pengalaman, agar dapat mengambil keputusan atau melakukan prediksi tanpa perlu diprogram secara eksplisit untuk setiap tugas.
21.	<i>Flowchart</i>	Sebuah diagram yang menggambarkan urutan langkah-langkah dalam suatu proses, dimulai dari tahap awal hingga tahap akhir secara sistematis.
22.	<i>Middleware</i>	Aplikasi yang berfungsi sebagai perantara antara dua sistem atau perangkat, sehingga memungkinkan keduanya untuk saling berkomunikasi dan bekerja secara terintegrasi.
23.	<i>Backend</i>	Bagian dari aplikasi yang berjalan di belakang layar dan bertanggung jawab atas pengelolaan data, logika proses, serta komunikasi antara antarmuka pengguna dan sistem inti.
24.	<i>Cloud</i>	Sistem penyimpanan dan pengolahan data yang dapat diakses melalui internet, sehingga pengguna tidak perlu menggunakan perangkat keras secara langsung di lokasi mereka.
25.	<i>Smart City</i>	Kota yang memanfaatkan teknologi untuk mengelola sumber daya dan layanan secara lebih efisien, dengan tujuan meningkatkan kenyamanan serta kualitas hidup masyarakat yang tinggal di dalamnya.

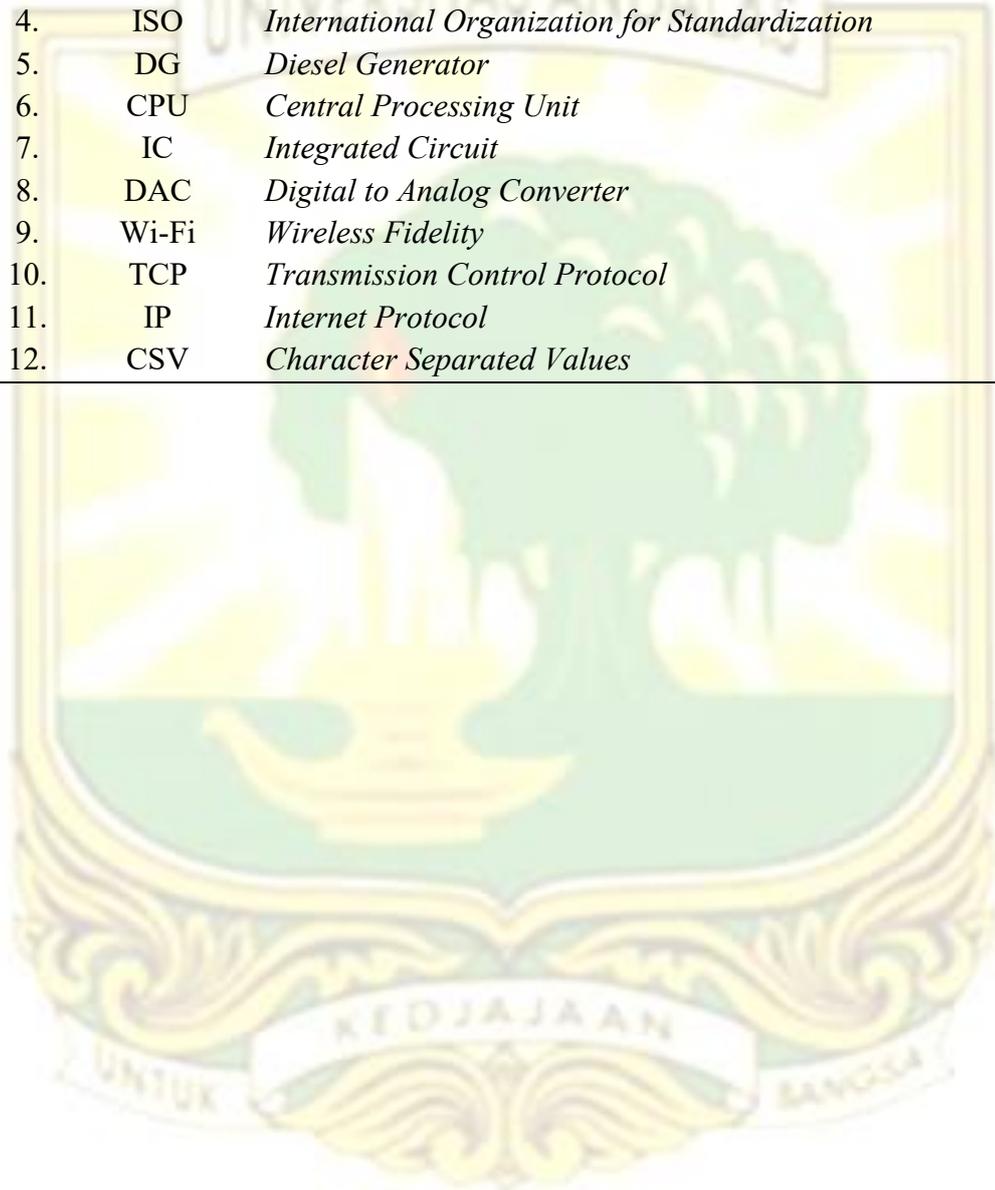
- ukuran seberapa banyak data yang dapat dikirim atau diterima melalui jaringan dalam jangka waktu tertentu, biasanya diukur dalam satuan bit per detik.
26. *Bandwidth*
- Semua bagian fisik dari perangkat elektronik yang bisa dilihat dan disentuh, seperti mesin, kabel, layar, atau komponen lainnya.
27. *Hardware*
- Bagian dari sistem komputer atau alat elektronik berupa program atau perintah yang mengatur cara kerja perangkat agar dapat menjalankan fungsinya dengan benar.
28. *Software*
- Istilah yang menggambarkan perangkat atau aplikasi yang bisa digunakan secara fleksibel di berbagai tempat tanpa perlu koneksi kabel tetap atau berada di satu lokasi.
29. *Mobile*
- Kondisi ketika sebuah model belajar terlalu keras dari data latihannya, sampai-sampai dia “hafal di luar kepala”. Akibatnya, model itu jadi jago banget sama data latihannya, tapi malah bingung atau tidak akurat waktu dikasih data baru yang belum pernah dia lihat sebelumnya.
30. *Overfitting*
- Sebuah modul kecil yang di dalamnya sudah terdapat berbagai komponen penting seperti pemroses, penyimpan data, dan pengatur kerja sistem, sehingga bisa menjalankan fungsi seperti sebuah komputer lengkap dalam satu modul.
31. *System on Chip*
- Hubungan antara dua besaran di mana perubahan pada salah satunya menyebabkan perubahan yang sebanding pada yang lainnya.
32. *Linearitas*
- Titik atau bagian kecil pada komponen elektronik yang berfungsi untuk menghubungkan, mengirim, atau menerima sinyal maupun arus listrik.
33. *Pin*
- Alat yang digunakan untuk mengukur percepatan atau gerakan suatu benda, baik dalam arah maju-mundur, naik-turun, maupun ke samping.
34. *Accelerometer*
- Alat yang digunakan untuk mengukur arah atau sudut putaran suatu benda disebut sensor sudut atau sensor posisi rotasi.
35. *Gyroscope*
- Nama atau identitas yang digunakan seseorang untuk masuk dan menggunakan suatu aplikasi.
36. *Username*
- Kode rahasia yang digunakan untuk menjaga keamanan agar hanya orang yang berhak yang bisa mengakses suatu sistem atau aplikasi.
37. *Password*
- Proses menyiapkan atau mengaktifkan suatu sistem, alat, atau variabel agar siap digunakan.
38. *Inisialisasi*
- Nilai atau pengaturan awal yang sudah ditentukan oleh sistem sebelum pengguna mengubahnya.
39. *Default*

40. *Clamp Meter* Alat ukur listrik yang digunakan untuk mengukur arus tanpa harus memutus atau menyentuh langsung kabel listrik.
41. *Broker* Aplikasi yang mengatur pengiriman dan penerimaan pesan antar perangkat dalam sebuah sistem.
42. *Pipeline* Serangkaian langkah sistematis yang dilakukan secara berurutan untuk mengolah data dari tahap awal hingga menghasilkan hasil akhir.
43. *Stream* Proses pengiriman atau pemrosesan data secara berkelanjutan dan langsung seiring data diterima, tanpa harus menunggu seluruh data terkumpul terlebih dahulu.
44. *Cloud Deployment* Cara menggunakan aplikasi atau sistem melalui layanan berbasis internet, memungkinkan akses dari berbagai lokasi tanpa perlu menyimpannya secara lokal di satu perangkat saja.
- 



## DAFTAR SINGKATAN

No.	Singkatan	Kepanjangan
1.	DC	<i>Direct Current</i>
2.	AC	<i>Alternating Current</i>
3.	AVR	<i>Automatic Voltage Regulator</i>
4.	ISO	<i>International Organization for Standardization</i>
5.	DG	<i>Diesel Generator</i>
6.	CPU	<i>Central Processing Unit</i>
7.	IC	<i>Integrated Circuit</i>
8.	DAC	<i>Digital to Analog Converter</i>
9.	Wi-Fi	<i>Wireless Fidelity</i>
10.	TCP	<i>Transmission Control Protocol</i>
11.	IP	<i>Internet Protocol</i>
12.	CSV	<i>Character Separated Values</i>



## BAB I PENDAHULUAN

### 1.1. Latar Belakang

Pada saat ini, energi listrik merupakan keperluan utama bagi manusia, karena segala hal yang berhubungan dengan aktivitas manusia tidak dapat lepas dari energi listrik. Berdasarkan catatan data energi listrik tahun 2013 hingga 2021, jumlah konsumsi energi listrik di Indonesia terus mengalami peningkatan rata-rata sebesar 6,96%[1]. Kebutuhan akan energi listrik meliputi dukungan untuk aktivitas ekonomi, sosial, transportasi, pendidikan, dan lain sebagainya.

Generator diesel berbasis sistem kekuatan darurat dipasang di fasilitas-fasilitas yang berisiko kehilangan pasokan listrik utama. Mereka berfungsi secara otomatis untuk mengambil alih beban penting selama pemadaman dan digunakan sebagai sumber daya independen terhubung dengan bangunan beban kritis. Meskipun efektif dalam pemadaman singkat, keandalan dan konfigurasi generator menjadi kritis dalam pemadaman yang lebih lama. Kegagalan generator darurat dapat berdampak ekonomi, risiko masyarakat, fasilitas medis, dan keamanan nasional jika terjadi di instalasi militer[2].

Seiring dengan meningkatnya pemanfaatan sumber energi terbarukan, khususnya energi angin dan surya, pembangkit listrik berbahan bakar fosil kini lebih banyak beroperasi dalam rentang beban menengah hingga puncak guna menjaga kestabilan jaringan listrik. Operasi dalam kondisi tersebut menyebabkan frekuensi siklus beban yang lebih tinggi, yang pada akhirnya menimbulkan tekanan akibat perpaduan panas dan gaya mekanis pada komponen penting seperti generator. Kondisi ini berpotensi menyebabkan kegagalan mendadak, percepatan proses penuaan mesin, risiko kerusakan, gangguan produksi, serta kebutuhan akan pemeliharaan tak terjadwal. Oleh karena itu, pemantauan melalui analisis getaran dan suhu menjadi metode penting dalam menjaga kinerja dan keandalan mesin-mesin berputar.[3].

Pemantauan kondisi mesin secara kontinu sangat penting dalam menjaga keandalan dan kinerja generator. Pemantauan *online* yang baik dapat membantu mendeteksi perubahan dalam kondisi mesin yang dapat menunjukkan kemungkinan kerusakan atau kegagalan, sehingga tindakan pencegahan atau perbaikan dapat diambil sebelum terjadi masalah yang serius. Hal ini dapat membantu menghindari pemadaman yang tidak terduga dan memperpanjang umur pakai generator. Selain itu, pemantauan kondisi mesin juga dapat membantu mengoptimalkan penggunaan energi dan mengurangi konsumsi bahan bakar yang tidak perlu, sehingga dapat mengurangi biaya operasional dan dampak lingkungan dari operasi generator.

Penambahan perangkat pengukur seperti sensor daya, getaran, dan suhu agar dapat memberikan informasi yang sangat penting dalam memantau kinerja mesin. Ada berbagai jenis sensor yang dapat digunakan untuk mengukur parameter kunci yang perlu dipantau secara terus menerus. Data yang diterima dari sensor tersebut kemudian dapat dipresentasikan dalam berbagai plot atau grafik yang membantu dalam memahami kondisi mesin secara *real-time* melalui pemantauan *online*. Teknik pemrosesan sinyal juga digunakan untuk mengekstrak fitur khusus yang sensitif terhadap kegagalan pada sinyal sensor tersebut. Pada alat ini, analisis tersebut dilakukan melalui berbagai sensor seperti sensor daya, sensor suhu, dan sensor getar[4].

Dengan adanya teknologi IoT, mesin dapat terhubung ke internet dan mengumpulkan data dari berbagai sensor dan perangkat pemantau lainnya secara *real-time*. Data yang dikumpulkan kemudian dapat disimpan di *localhost* dan dianalisis dengan algoritma *machine learning* atau teknik pemrosesan data lainnya untuk mendapatkan informasi yang berguna tentang kinerja mesin, dan memperoleh peringatan dini jika terjadi masalah pada mesin. Dengan demikian, pengguna dapat mengambil tindakan preventif sebelum masalah menjadi lebih serius dan berdampak negatif pada operasional mesin. Antarmuka pengguna grafis yang baik juga dapat membantu pengguna untuk memahami data yang dianalisis dengan mudah dan cepat. Untuk mengatasi permasalahan yang ada, diperlukan peramalan jangka pendek untuk melakukan prediksi perawatan dari genset. Prediksi ini didasarkan pada data daya, suhu, dan getaran yang dihasilkan oleh genset selama 24 jam. Penelitian ini menggunakan model peramalan yang menggunakan data masa lalu dan sekarang. Tujuan utama dari menggunakan teknologi CNN adalah untuk mengidentifikasi data dari sensor yang terpasang pada genset, sehingga dapat menunjukkan tanda-tanda awal kerusakan pada peralatan tersebut.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang penulisan tugas akhir yang telah diuraikan di atas, maka rumusan masalah pada tugas akhir ini dapat dirumuskan sebagai berikut:

1. Bagaimana merancang sistem *monitoring* genset?
2. Bagaimana proses pengiriman data *real-time* hasil *monitoring* genset?
3. Bagaimana proses penyimpanan data hasil *monitoring* ke *localhost*?
4. Bagaimana proses implementasi data untuk peramalan menggunakan metode CNN?
5. Bagaimana hasil analisa tingkat kerusakan genset menggunakan metode CNN?

### 1.3. Tujuan Penelitian

Adapun tujuan dari tugas akhir ini, sebagai berikut :

1. Merancang sistem *monitoring* kinerja genset berbasis *Internet of Things*.
2. Menyimpan data hasil *monitoring* di *database* untuk proses peramalan tingkat kerusakan genset.
3. Menganalisis alur kerja algoritma peramalan tingkat kerusakan genset menggunakan algoritma CNN.

### 1.4. Manfaat Penelitian

Hasil Penelitian dan penulisan tugas akhir ini diharapkan dapat memberikan manfaat berupa:

1. Mengetahui daya, suhu, dan getaran dari genset secara *real-time* menggunakan internet.
2. Mengetahui tingkat kerusakan dari genset.
3. Membuat perkiraan daya, suhu, dan getaran yang dihasilkan oleh genset berdasarkan data terukur dengan metode CNN.

### 1.5. Batasan Masalah

Adapun batasan masalah pada penelitian ini yaitu :

1. Penelitian ini di uji pada Genset Pesantren Harakatul Quran, Koto Tangah, Kota Padang.
2. Parameter yang digunakan pada alat untuk *me-monitoring* hanya menampilkan tegangan, arus, daya, getaran dan suhu dari hasil pengukuran sensor-sensor pada genset.
3. *Monitoring* data berbasis *Internet of Things* dengan akuisi dan transmisi data menggunakan ESP32.
4. Proses peramalan dilakukan secara manual dengan mengakses data histori sistem *monitoring* pada *database*.
5. Peramalan daya jangka pendek berdasarkan metode *Convolutional Neural Network* dengan *input* daya, suhu dan getaran.
6. Data histori yang digunakan untuk data latih peramalan jangka pendek dalam kurun waktu 3 hari.

### 1.6. Sistem Penulisan

Dalam laporan tugas akhir ini yang terbagi menjadi beberapa bab dengan sistematika tertentu, sistematika laporan ini adalah sebagai berikut.

#### BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang dari masalah dalam pembuatan tugas akhir ini, tujuan yang ingin dicapai, manfaat penelitian, batasan masalah, dan sistem penulisan.

## BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang teori-teori pendukung yang digunakan untuk memecahkan masalah dalam tugas akhir ini.

## BAB III METODOLOGI PENELITIAN

Bab ini memuat informasi tentang metodologi penelitian yang digunakan berupa metode penelitian, *flowchart* penelitian, peralatan dan bahan yang akan digunakan.

## BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi informasi tentang hasil dan pembahasan mengenai hasil penelitian.

## BAB V PENUTUP

Bab ini berisi kesimpulan tentang hasil penelitian dan saran untuk penelitian berikutnya yang menggunakan referensi penelitian ini.

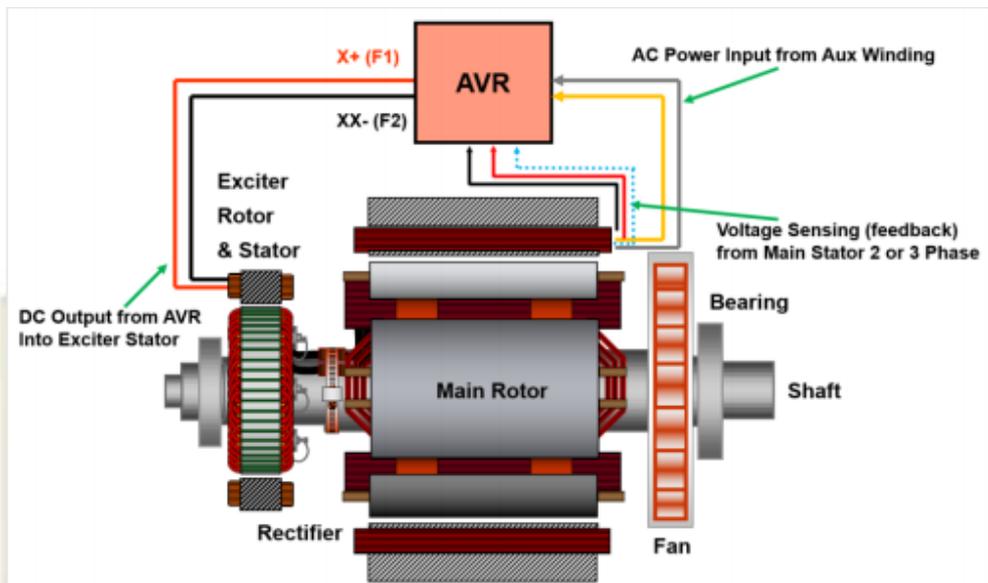


## BAB II TINJAUAN PUSTAKA

### 2.1. *Diesel Generator Set*

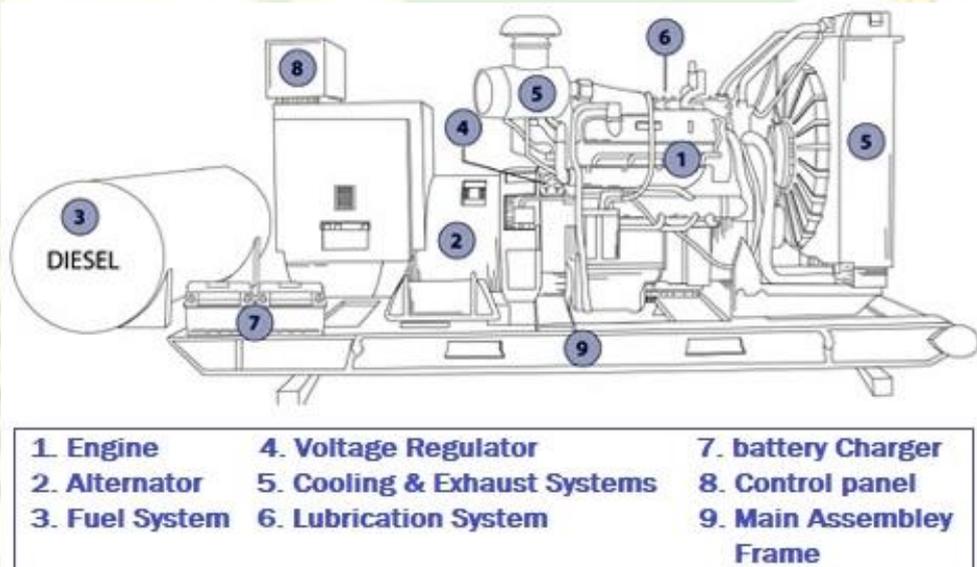
*Diesel Generator Set* atau biasa di kenal dengan genset, adalah sebuah perangkat yang terdiri dari generator dan mesin penggerak yang bekerja bersama untuk menghasilkan tenaga listrik. Mesin penggerak pada genset biasanya berupa motor diesel atau bensin, yang menggunakan bahan bakar seperti solar. Generator, di sisi lain, berfungsi untuk mengubah energi mekanik menjadi energi listrik. Prinsip kerja generator didasarkan pada eksperimen Faraday, di mana putaran magnet dalam kumparan atau sebaliknya menghasilkan perubahan fluks magnetik di dalam kumparan. Hal ini menyebabkan terjadinya beda potensial antara ujung-ujung kumparan, yang menghasilkan listrik[5].

Generator diesel bekerja berdasarkan prinsip konversi energi menurut hukum termodinamika. Prosesnya meliputi pembakaran bahan bakar, yang mengubah energi kimia menjadi energi mekanik rotasi. Energi mekanik ini digunakan untuk memutar rotor alternator, yang menghasilkan listrik AC. Listrik AC tersebut kemudian dikonversi menjadi listrik DC menggunakan penyearah. Rotor alternator yang dikelilingi oleh magnet berputar di dalam stator yang terbuat dari kawat tembaga. Pergerakan magnet di sepanjang kawat tembaga menghasilkan medan magnet yang menghasilkan tegangan listrik dalam rangkaian. Sistem *excitation* yang menggunakan arus searah (DC) digunakan untuk menciptakan medan elektromagnetik pada rotor. Listrik yang dihasilkan oleh alternator kemudian dikirimkan ke lokasi yang diinginkan dengan bantuan regulator tegangan[6]. Untuk diagram konversi energinya bisa dilihat pada Gambar 2.1 di bawah ini.



**Gambar 2. 1** Diagram Konversi pada Diesel Generator Set[6]

Pada ilustrasi yang diberikan dalam Gambar 2.2, ditampilkan berbagai komponen yang terdapat dalam sebuah generator set. Komponen-komponen tersebut meliputi:



**Gambar 2. 2** Komponen - Komponen Genset[7]

Berikut ini adalah beberapa kriteria kinerja yang harus dipenuhi oleh genset[7]

1. Tingkat Adaptasi terhadap Sistem Tenaga Listrik

Karakteristik dan kinerja genset sangat dipengaruhi oleh perannya dalam sistem tenaga listrik. Pada jaringan listrik yang terisolasi, perubahan beban dan kecepatan mesin memiliki pengaruh yang lebih signifikan. Oleh karena itu, aplikasi ini bertujuan untuk menekan biaya energi dan menjaga parameter keluaran tetap dalam batas yang ditentukan. Pengelolaan sistem semacam ini membutuhkan pemahaman mendalam terhadap karakteristik beban dan batasan perangkat sakelar. Dalam penelitian ini, genset dihubungkan ke bus DC melalui perangkat pengubah daya, dengan kecepatan rotor yang disesuaikan berdasarkan permintaan daya dari sisi bus DC. Perubahan beban serta kondisi pengisian baterai menjadi faktor utama dalam pengendalian parameter sistem.

2. Kapasitas Bahan Bakar Diesel dan Daya Tersedia

Desain dan ukuran generator diesel ditentukan oleh kebutuhan daya, kondisi operasional, dan lingkungan yang ada. Beban listrik yang harus ditangani dan produksi energi terbarukan mempengaruhi performa generator diesel. Fluktuasi beban dan pertumbuhan beban dari tahun ke tahun merupakan faktor penting dalam menentukan desain dan ukuran generator diesel. Pada jaringan listrik mandiri, fluktuasi beban cenderung tinggi, dan menggunakan beberapa generator diesel dengan kapasitas yang lebih rendah dapat meningkatkan efisiensi sistem dan mengurangi biaya pemeliharaan.

3. Efisiensi

Parameter seperti karakteristik beban, kualitas bahan bakar, kondisi lingkungan, desain dan kapasitas mesin, kelembaban, dan jam operasi memengaruhi efisiensi generator diesel. Fluktuasi beban yang rendah atau tinggi dapat mengurangi efisiensi konsumsi bahan bakar dan kualitas daya listrik yang dihasilkan. Standar dan pedoman telah ditetapkan untuk mengukur efisiensi generator diesel dan membatasi operasionalnya. Studi telah dilakukan untuk mengestimasi cara meningkatkan performa generator diesel. Penggunaan simulasi komputer dengan memperhitungkan profil beban dapat membantu meningkatkan efisiensi dengan memperkirakan konsumsi bahan bakar. Efisiensi generator diesel bergantung pada efisiensi mesin dan alternator serta faktor lingkungan selama operasionalnya. Rumus yang digunakan untuk menghitung efisiensi generator diesel ditunjukkan pada rumus (2.1), di bawah ini.

$$\eta = \frac{P}{E} \quad (2.1)[7]$$

Keterangan :

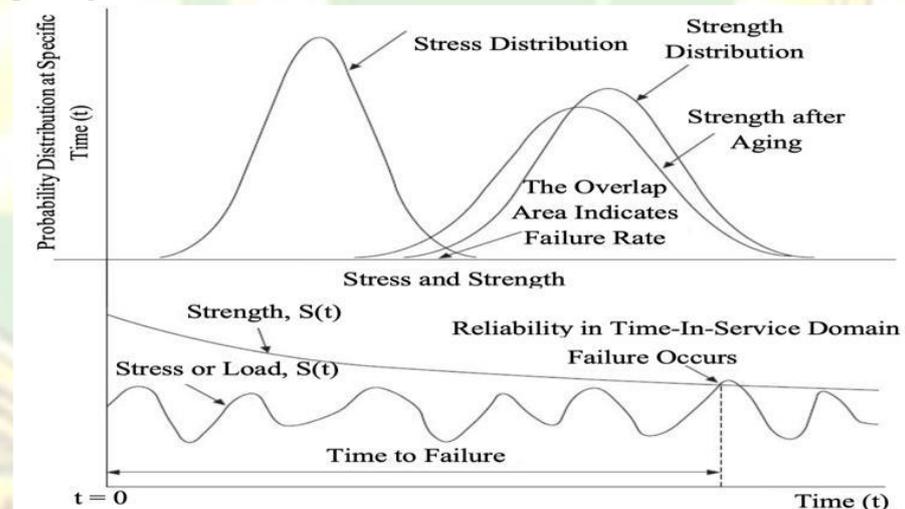
$\eta$  : Efisiensi Genset

P : Daya yang Dihasilkan

E : Energi Pembakaran Bahan Bakar

#### 4. Daya Tahan

Ketahanan atau daya tahan merupakan faktor penting yang mempengaruhi kelayakan, ketersediaan, dan biaya perawatan peralatan dalam berbagai kondisi operasi. Mesin dan generator listrik pada sistem generator diesel harus mampu menahan tekanan mekanis, suhu tinggi, dan kerusakan struktural. Tes ketahanan dilakukan untuk memeriksa batas toleransi komponen mesin. Penggunaan yang berlebihan atau beban tak terduga dapat meningkatkan suhu dan merusak sistem isolasi generator. Diagram interferensi antara beban dan kekuatan, seperti yang ditunjukkan pada Gambar 2.3, mempengaruhi ketahanan dan keandalan generator diesel. Beban mencerminkan tekanan yang diberikan pada sistem, sedangkan kekuatan mengacu pada kemampuan struktural perangkat keras.



**Gambar 2.3** Distribusi Probabilitas Acak dari Kekuatan dan Tekanan Stres[7]

5. Spesifikasi Genset

Standar ISO 8528-1 untuk generator diesel menetapkan spesifikasi yang tepat untuk memastikan kualitas daya yang aman dan efisien. Standar ini mencakup empat klasifikasi daya yang berbeda, seperti Daya Cadangan Darurat dan Daya Operasi Berkelanjutan, yang memberikan panduan kepada produsen dan konsumen tentang batasan sistem yang diterapkan. Spesifikasi ini mencakup berbagai faktor penting, termasuk beban maksimum dan minimum, harmonik, faktor daya, dan kualitas bahan bakar. Dengan adanya standar ini, diharapkan pelayanan kepada pelanggan dapat terjamin dan generator diesel dapat beroperasi dengan baik sesuai dengan kebutuhan yang ditentukan.

6. Sistem *Monitoring* dan Pengontrolan Genset

Sistem *Automation* Genset memiliki peran penting dalam memantau produksi dan menganalisis kondisi sistem. Sistem ini juga dikenal sebagai sistem otomatisasi-*grid* yang menggunakan metode kontrol canggih untuk mengatur frekuensi dan tegangan. Estimasi dan penyesuaian kecepatan menjadi kunci dalam kontrol kecepatan variabel. Metode mekanik atau matematis digunakan untuk mengevaluasi posisi dan kecepatan rotor. Tujuan dari *monitoring* dan pengontrolan ini adalah menghubungkan kecepatan mesin diesel dengan variasi beban dan menyesuaikan torsi listrik yang dihasilkan oleh generator tenaga angin. Sistem manajemen energi yang diusulkan juga menyediakan fungsi start dan stop otomatis untuk melindungi sistem dari beban yang berlebihan.

7. Harga Bahan Bakar

Biaya bahan bakar Genset tergantung pada konsumsi bahan bakar, harga bahan bakar, dan jam operasi. Konsumsi bahan bakar dipengaruhi oleh beban, efisiensi genset, dan kualitas bahan bakar. Harga bahan bakar adalah biaya per volume atau unit yang dibeli. Jam operasi adalah durasi waktu operasional Genset. Dengan menghitung biaya bahan bakar, pemilik Genset dapat memperkirakan biaya operasional berdasarkan konsumsi bahan bakar dan faktor-faktor terkait.

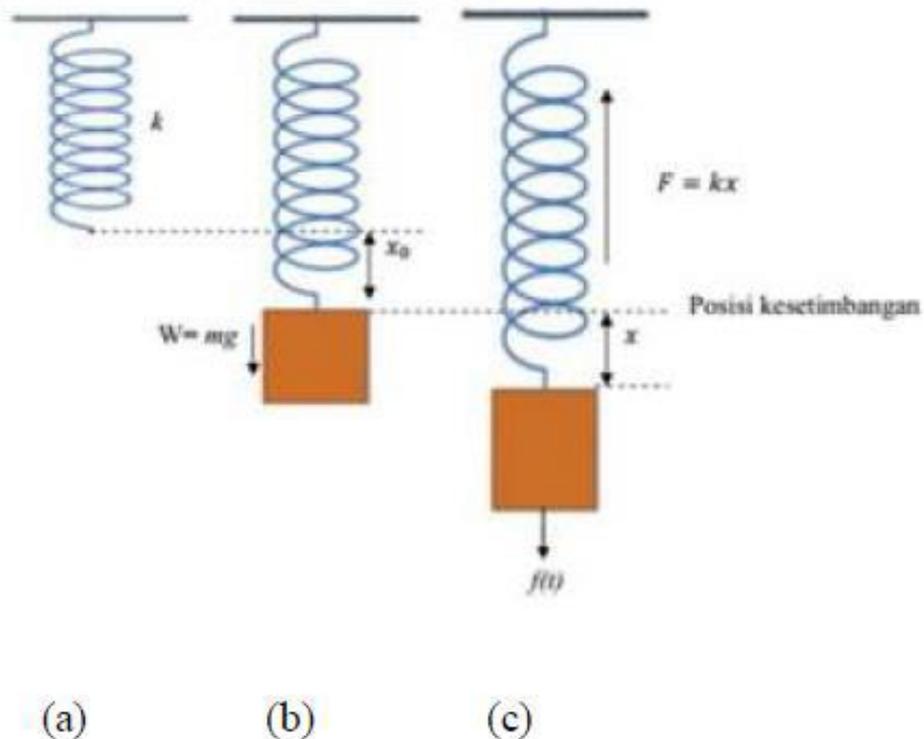
8. Daya Keluaran

Produksi daya listrik oleh DG tergantung pada kapasitas maksimum dan minimum yang ditentukan oleh produsen. Faktor-faktor seperti kecepatan rotasi rotor, impedansi beban, dan sistem eksitasi rotor juga mempengaruhi keluaran generator. Daya yang dihasilkan oleh DG dapat dihitung dan didefinisikan dengan menggunakan parameter dan persamaan yang telah ditentukan dalam standar referensi yang relevan.

## 2.2. Getaran

Getaran adalah gerakan bolak-balik dengan amplitudo yang sama. Komponen penting dari getaran meliputi frekuensi, yang mengukur berapa kali gerakan bolak-balik terjadi dalam satu unit waktu, dan amplitudo, yang mengukur sejauh mana objek bergerak dari posisi pusatnya. Jenis getaran yang umum meliputi[8]:

1. Gerak Harmonik Sederhana: Ini adalah jenis getaran di mana objek bergerak bolak-balik dengan frekuensi tetap dan amplitudo yang konstan. Contoh dari gerak harmonik sederhana adalah ayunan bandul matematis.
2. Gerak Harmonik Tersedam: Dalam jenis getaran ini, gerakan bolak-balik juga terjadi dengan frekuensi tetap, tetapi amplitudo secara bertahap berkurang seiring waktu. Ini terjadi karena adanya gaya gesekan atau hambatan lainnya. Contoh dari gerak harmonik teredam adalah peredaman getaran dalam suspensi kendaraan.
3. Gerak Harmonik Terdampak: Dalam jenis ini, terdapat pengaruh eksternal yang memengaruhi gerakan bolak-balik. Misalnya, pemberian dorongan tambahan pada objek yang bergetar. Ini menciptakan perubahan dalam frekuensi dan amplitudo getaran.



Gambar 2. 4 Pegas Vertikal[8]

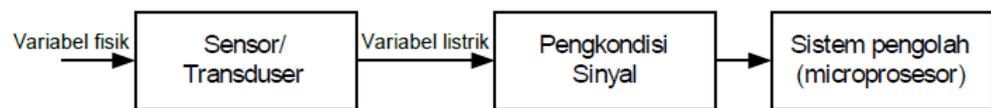
Gambar 2.5, yang mengilustrasikan berbagai kondisi pegas tergantung yang memengaruhi getaran. Standar pengukuran getaran yang sering digunakan adalah ISO 10816-3:1998(E) yang ditampilkan pada tabel 2.1, yang berisi pedoman untuk menilai getaran mesin berputar. Ini mencakup metode pengukuran, klasifikasi tingkat getaran, dan kriteria evaluasi kesehatan mesin berdasarkan analisis getaran[9].

**Tabel 2. 1** Kriteria Zona Evaluasi Kelayakan Getaran[9]

<b>VIBRATION SEVERITY PER ISO 10816</b>						
	<b>Machine</b>		<b>Class I</b>	<b>Class II</b>	<b>Class III</b>	<b>Class IV</b>
	<b>In/s</b>	<b>mm/s</b>	<b>small Machine</b>	<b>medium Machine</b>	<b>large grid foundation</b>	<b>large soft foundation</b>
<b>Vibration Velocity Vrms</b>	0.01	0.28				
	0.02	0.45				
	0.03	0.71				
	0.04	1.12				
	0.07	1.80				
	0.11	2.80	satisfactory			
	0.18	4.50				
	0.28	7.10		unstatsifactory		
	0.44	11.2				
	0.70	18.0				
	0.71	28.0	unacceptable			
	1.10	45.0				

### 2.3. Sensor

Sensor dan transduser adalah perangkat elektronik yang mengubah besaran fisik menjadi besaran listrik, ilustrasinya dapat dilihat pada Gambar 2.6. Mereka digunakan dalam berbagai peralatan modern dan memiliki ukuran yang semakin kecil. Sensor dapat dibedakan menjadi dua jenis, yaitu sensor fisika yang mendeteksi besaran fisik seperti cahaya, suara, gaya, kecepatan, dan suhu, serta sensor kimia yang mendeteksi jumlah zat kimia melalui reaksi kimia. Sensor digunakan dalam berbagai bidang seperti otomotif, kedokteran, industri, robotika, dan *aerospace*. Besaran listrik itu sendiri berupa tegangan, resistansi dan arus listrik. Dalam sistem kontrol dan robotika, sensor berfungsi seperti indra yang memberikan data kepada pengontrol untuk diolah[10].



**Gambar 2. 5** Blok Fungsional Sensor/*Tranduser*[10]

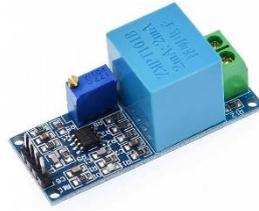
Sensor itu sendiri terbagi atas beberapa jenis, yaitu: Sensor *Proximity*, Sensor Magnet, Sensor Sinar, Sensor Ultrasonik, Sensor Tekanan, Sensor Kecepatan, Sensor *Encoder*, Sensor Suhu, Sensor *Hall Effect*, Sensor Arus Sensor Tegangan, Sensor Getaran, dll[10].

### 2.3.1. Sensor Tegangan (ZMPT101B)

Modul sensor tegangan ZMPT101B merupakan perangkat pengukur tegangan AC yang menggunakan transformator tegangan sebagai komponen utamanya. Sensor ini mampu mengukur tegangan hingga 250V AC dengan tingkat akurasi dan konsistensi yang tinggi. Salah satu keunggulan dari sensor ini adalah adanya penyetel presisi, yang memungkinkan pengguna untuk menyesuaikan keluaran ADC (*Analog-to-Digital Converter*) agar sesuai dengan nilai tegangan referensi. Fitur ini sangat membantu dalam proses kalibrasi, sehingga mempermudah penggunaannya dalam berbagai aplikasi pemantauan tegangan berbasis *mikrokontroler* [11].

Dalam penelitian ini, proses kalibrasi sensor dilakukan dengan membandingkan nilai tegangan efektif ( $V_{rms}$ ) yang diperoleh dari pembacaan sensor dengan hasil pengukuran menggunakan voltmeter standar. Untuk memperkuat validitas hasil, dilakukan pula analisis regresi guna menemukan hubungan matematis yang lebih akurat antara tegangan *input* dan *output* ADC. Kombinasi antara akurasi tinggi, kemudahan dalam kalibrasi, serta kemampuan dalam pemantauan secara *real-time* menjadikan sensor ZMPT101B sebagai solusi yang andal dalam sistem *monitoring* tegangan[12].

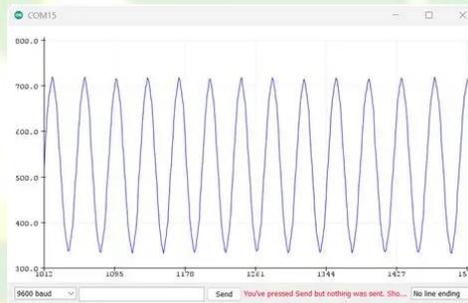
Gambar 3.3 memperlihatkan bentuk fisik dari modul ZMPT101B. Sementara itu, karakteristik *output* sensor terhadap tegangan *input* divisualisasikan dalam Gambar 3.4, yang menunjukkan grafik respons sensor. Untuk informasi lebih detail terkait kemampuan teknis dan parameter kerja dari sensor ini, dapat dilihat pada Tabel 3.2 yang mencantumkan spesifikasi teknis ZMPT101B.



Gambar 3. 1 ZMPT101B[12]

Tabel 3. 1 Spesifikasi ZMPT101B[12]

No.	Fitur	Spesifikasi
1.	Tegangan <i>Input</i>	5-30 V DC
2.	Sinyal <i>Output</i>	0-5 V
3.	Rentang Tegangan Terdeteksi	0-250 V AC
4.	Arus <i>Input</i> Maksimum	2 mA
5.	Dimensi	49,5 mm x 19,4 mm
6.	Rentang Suhu Operasional	40-70 °C



Gambar 3. 2 Grafik Respons ZMPT101B[12]

Grafik respons pada Gambar 3.4 menunjukkan hasil pembacaan tegangan oleh sensor ZMPT101B yang divisualisasikan melalui serial monitor pada Arduino IDE. Grafik ini merepresentasikan bagaimana sensor merespons perubahan tegangan *input* dalam bentuk data digital yang dikonversi oleh ADC (*Analog to Digital Converter*) dan dikirim ke komputer melalui komunikasi serial. Pola grafik yang dihasilkan mencerminkan kestabilan, sensitivitas, serta keakuratan pembacaan sensor terhadap sinyal AC yang diukur. Melalui tampilan ini, pengguna dapat memantau fluktuasi tegangan secara *real-time* dan melakukan analisis terhadap performa sensor dalam kondisi pengujian tertentu.

### 2.3.2. Sensor Arus (ACS712)

Sensor ACS712 adalah sensor arus berbasis efek *Hall* yang mampu mengukur arus searah (DC) maupun bolak-balik (AC) secara akurat. Sensor ini bekerja dengan mendeteksi medan magnet yang dihasilkan oleh arus yang mengalir melalui jalur internal sensor. Medan magnet tersebut kemudian diterjemahkan oleh elemen *Hall* menjadi tegangan yang proporsional terhadap besarnya arus, dan sinyal ini dapat dibaca oleh *mikrokontroler* melalui *pin* analog. Karena kemampuannya dalam memberikan hasil pengukuran yang stabil dan *real-time*, sensor ini banyak digunakan dalam aplikasi seperti sistem proteksi beban, kontrol motor, serta pemantauan konsumsi daya listrik [13].

Sensor ACS712 tersedia dalam tiga varian berdasarkan kapasitas pengukuran arus, yaitu 5 A, 20 A, dan 30 A. Dalam penelitian ini digunakan varian 30 A karena cocok untuk sistem dengan beban arus yang relatif tinggi. Meskipun sensor ini cukup andal dan mudah diintegrasikan dengan *mikrokontroler*, terdapat satu kelemahan yang perlu diperhatikan, yaitu metode pemasangannya. Untuk dapat mengukur arus, sensor ini harus dipasang secara seri dengan kabel yang akan diukur, yang berarti pengguna perlu memutus jalur kabel daya terlebih dahulu sebelum menyambungkannya ke *input* sensor. Hal ini membuat proses instalasi memerlukan perhatian khusus, terutama pada sistem yang sudah beroperasi [14].

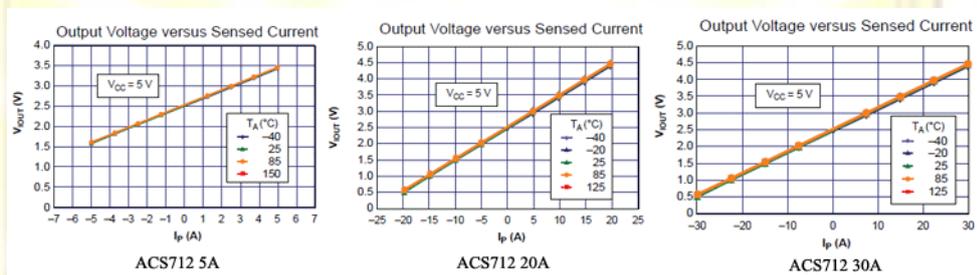
Gambar 3.5 menampilkan tampilan fisik dari modul ACS712. Sementara itu, hubungan antara arus *input* dan *output* sensor divisualisasikan melalui grafik respons yang ditunjukkan pada Gambar 3.6. Informasi lebih lanjut mengenai performa teknis dan parameter operasional dari sensor ini dapat ditemukan pada Tabel 3.3, yang memuat spesifikasi teknis ACS712.



**Gambar 3. 3** ACS712[14]

Tabel 3. 2 Spesifikasi ACS712[14]

No.	Fitur	Spesifikasi
1.	Arus Primer	2 mA
2.	Arus Sekunder	2 mA
3.	Eror Sudut Fasa	< 20° (50 Ohm)
4.	Jangkauan Arus	0-30 A
5.	Linearitas	0.1 %
6.	Akurasi	0,2
7.	Level Dielektrik	3000 VAC/Min
8.	Resistansi DC pada Suhu 20 °C	110 Ohm

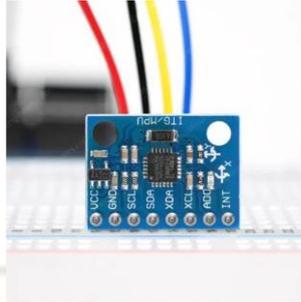


Gambar 3. 4 Grafik Respons ACS712[14]

Gambar 3.6 menggambarkan kurva respons sensor ACS712 terhadap arus masukan ( $I_p$ ) dengan tegangan *output* ( $V_{out}$ ) untuk tiga varian, yaitu 5A, 20A, dan 30A. Ketiga varian menunjukkan pola hubungan linier antara arus yang terukur dan tegangan keluaran, di mana titik nol arus (0 A) berada di sekitar 2,5V ketika diberi catu daya 5V ( $V_{cc}$ ). Meskipun sensor diuji dalam rentang suhu operasi antara -40°C hingga 150°C, hasil pengukuran tetap stabil dan konsisten. Varian dengan kapasitas arus yang lebih tinggi cenderung memiliki sensitivitas yang lebih rendah, namun memungkinkan pengukuran arus dalam rentang yang lebih luas.

### 2.3.3. Sensor Getaran dan Suhu (MPU6050)

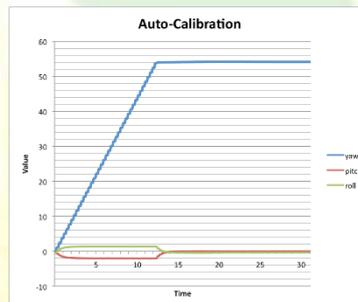
Modul MP6050 adalah sebuah sensor yang menggunakan teknologi *Micro Electro Mechanical Systems* (MEMS) dan memiliki fungsi untuk mengukur berbagai parameter terkait gerakan suatu objek. Sensor ini terdiri dari *accelerometer* 3 sumbu dan *gyroscope* 3 sumbu yang dapat mendeteksi percepatan, perpindahan, orientasi, kecepatan, dan parameter gerakan lainnya. Ketika terjadi perubahan gerakan, sensor ini mengubah perubahan tersebut menjadi variasi tegangan yang menghasilkan perubahan nilai-nilai yang diukur[15]. Gambar 3.7 menampilkan bentuk MPU6050, sementara Tabel 3.4 merangkum beberapa spesifikasinya.



**Gambar 3. 5** MPU6050[15]

**Tabel 3. 3** Spesifikasi MPU6050[15]

No.	Fitur	Spesifikasi
1.	Tegangan <i>Input</i>	2,375-3,46V DC
2.	Arus <i>Input</i> Maksimal	3,9mA
3.	Toleransi Kalibrasi	±3%
4.	Rentang Pengukuran <i>Accelerometer</i>	±2g, ±4g, ±8g, ±16g
5.	Rentang Pengukuran <i>Gyroscope</i>	±250°/sar, ±500°/sar, ±1000°/sar, ±2000°/sar



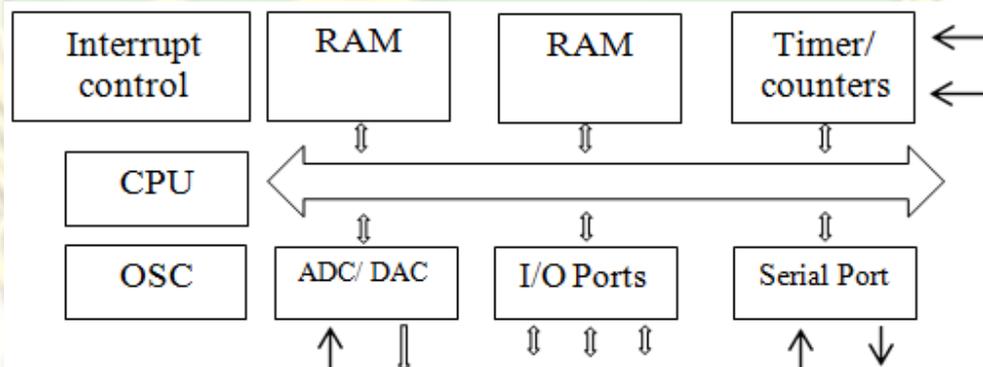
**Gambar 3. 6** Grafik Respons MPU6050[15]

Grafik respons MPU6050 pada Gambar 3.8 menunjukkan perubahan sudut *yaw*, *pitch*, dan *roll*. *Yaw* adalah rotasi pada sumbu vertikal, *pitch* adalah kemiringan ke depan atau belakang, dan *roll* adalah kemiringan ke kiri atau kanan. Grafik ini membantu menganalisis pergerakan dan orientasi sensor dalam tiga dimensi.

## 2.4. Mikrokontroler

*Mikrokontroler* merupakan sebuah perangkat komputer kecil yang terdiri dari CPU, memori, *timer*, saluran komunikasi, dan *port input/output*. Fungsinya adalah untuk menjalankan program tertentu dalam tugas-tugas yang diberikan. Penggunaan *mikrokontroler* sangat umum dalam kehidupan sehari-hari, seperti pada jam digital, televisi, sistem keamanan rumah, dan lainnya. Selain itu, *mikrokontroler* juga banyak digunakan dalam penelitian dan pengembangan, serta hadir dalam berbagai sistem elektronika modern seperti kendaraan, peralatan rumah tangga, industri, telekomunikasi, dan lain-lain. Keuntungan menggunakan *mikrokontroler* meliputi harga yang terjangkau, kemampuan untuk diprogram ulang, dan fleksibilitas dalam penyesuaian dengan kebutuhan. Saat ini, terdapat beberapa keluarga *mikrokontroler* yang populer, seperti Intel MCS51, Motorola 68HC11, Microchip PI, Hitachi H8, dan Atmel AVR.

Sejarah *mikrokontroler* dimulai dengan ditemukannya mikroprosesor dan komputer. Kemudian, perkembangan berlanjut dengan penemuan transistor, IC, dan mikroprosesor seperti Intel 4004 dan 8008. Pada tahun 1972, Texas Instrument menciptakan *mikrokontroler* TMS 1000, diikuti oleh Intel 8048 pada tahun 1976. Mikroprosesor 16 bit seperti Intel 8086 juga mulai populer. Pada tahun 1980, Intel 8051 menjadi *mikrokontroler* yang terkenal, diikuti oleh Atmel AVR pada tahun 1996 dengan fitur penyimpanan program *on-chip flash*. Sejak saat itu, *mikrokontroler* terus mengalami perkembangan dengan fungsionalitas canggih dan aplikasi yang semakin luas. Beberapa fitur yang umumnya hadir dalam *mikrokontroler*, ditunjukkan pada Gambar 2.7.



Gambar 2. 6 Alur Proses Mikrokontroler[16]

Gambar 2.7 adalah ilustrasi fitur-fitur yang terdapat pada *mikrokontroler*. Berikut adalah penjelasan singkat tentang setiap fitur[16]:

1. RAM (*Random Access Memory*)

RAM (*Random Access Memory*) adalah area penyimpanan variabel yang digunakan untuk menyimpan data sementara. Namun, data di RAM akan hilang jika tidak ada catu daya.

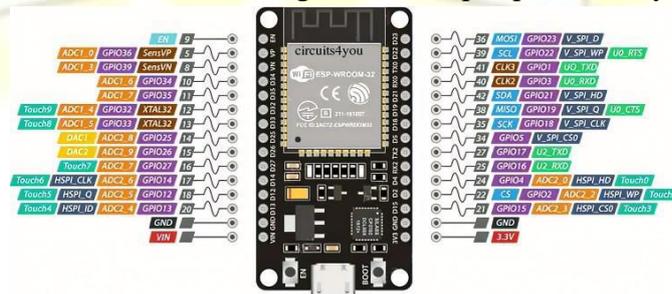
2. ROM (*Read Only Memory*)  
ROM (*Read Only Memory*) adalah tempat penyimpanan program yang disediakan oleh *programmer*. Biasanya disebut sebagai "memori kode" karena isinya tidak dapat diubah.
3. Register  
Register adalah tempat penyimpanan nilai-nilai yang digunakan dalam proses. Data yang disimpan di register bersifat sementara.
4. SFR (*Special Function Register*)  
SFR (*Special Function Register*) adalah register khusus yang mengatur operasi *mikrokontroler*. SFR terletak di dalam RAM.
5. *Input dan Output (I/O) Pin*  
*Input* dan *Output pin* berfungsi sebagai penerima dan penghasil sinyal. *Input pin* digunakan untuk menerima sinyal dari luar, seperti sensor, sedangkan *output pin* menghasilkan sinyal berdasarkan proses algoritma *mikrokontroler*.
6. *Interrupt*  
*Interrupt* (intrusi) adalah mekanisme yang memungkinkan program utama terganggu secara internal. Hal ini berguna ketika perlu menangani kejadian mendesak saat program utama sedang berjalan.
7. *External Interrupt*  
*External Interrupt* adalah jenis intrusi yang terjadi ketika ada *input* dari *pin* interupsi di luar *mikrokontroler*.
8. *Interrupt Timer*  
*Interrupt Timer* adalah intrusi yang terjadi pada waktu tertentu sesuai dengan pengaturan waktu yang ditentukan. Contohnya, digunakan untuk penundaan dalam satuan *milidetik*.
9. *Interrupt Serial*  
*Interrupt Serial* terjadi saat menerima data selama komunikasi serial atau ketika register penuh selama proses penerimaan data serial dari luar.

#### 2.4.1. ESP32

ESP32 adalah sebuah *mikrokontroler System on Chip (SOC)* yang memiliki harga terjangkau dan konsumsi daya rendah. SoC ini memiliki fitur terintegrasi Wi-Fi dan Bluetooth dual-mode, serta mendukung teknologi daya rendah. Semua fitur ini dikemas dalam satu *chip*, menjadikan ESP32 sebagai solusi yang efisien dan serbaguna untuk berbagai aplikasi yang membutuhkan konektivitas nirkabel dan pengendalian *mikrokontroler*[17].

Sistem ini dirancang dengan menggunakan modul ESP32 DEVKIT V1 yang dilengkapi dengan beberapa *pin* GPIO, dua pasang *pin* komunikasi serial, dan RTC. Modul perangkat keras ini juga dapat menyediakan pasokan daya 5V dan 3V untuk modul terintegrasi lainnya[18].

Salah satu keuntungan utama dari penggunaan ESP32 adalah kemampuannya untuk dikodekan menggunakan berbagai platform dan bahasa sumber terbuka, seperti Arduino IDE, dan langsung diunggah ke papan. Gambar 3.2 menampilkan gambar papan ESP32 DEVKIT V1, sementara Tabel 3.1 merangkum beberapa spesifikasinya.



**Gambar 3. 7** ESP32 DEVKIT V1[18]

**Tabel 3. 4** Spesifikasi ESP32 DEVKIT V1[18]

No.	Fitur	Spesifikasi
1.	CPU	Tensilica Xtensa LX6 32bit Dual-Core di 160/240MHz
2.	SRAM	520 Kb
3.	FLASH	4 MB
4.	Tegangan Operasi	3.3 V
5.	Tegangan <i>Input</i>	7-12V
6.	Wi-Fi	802.11 b/g/n/e/i
7.	PIN	43
	UART	3
	GPIO	25
	SPI	4
	I2C	3
	ADC	6
	DAC	2

### 2.4.2. Arduino IDE

IDE (*Integrated Development Environment*) merupakan aplikasi yang digunakan untuk mengembangkan program pada ESP32 DEVKIT V1. Program yang dibuat menggunakan Arduino IDE disebut sebagai sketsa. Sketsa tersebut ditulis menggunakan editor teks dan disimpan dengan ekstensi *file* “.ino”. Arduino IDE juga memiliki sebuah kotak pesan hitam yang menampilkan informasi seperti pesan kesalahan, hasil kompilasi program, dan proses pengunduhan. Di bawah Arduino IDE, di sudut kanan, terdapat sebuah kartu yang menampilkan *port* COM aktif[19]. Arduino IDE memiliki tiga komponen utama[20], yaitu:

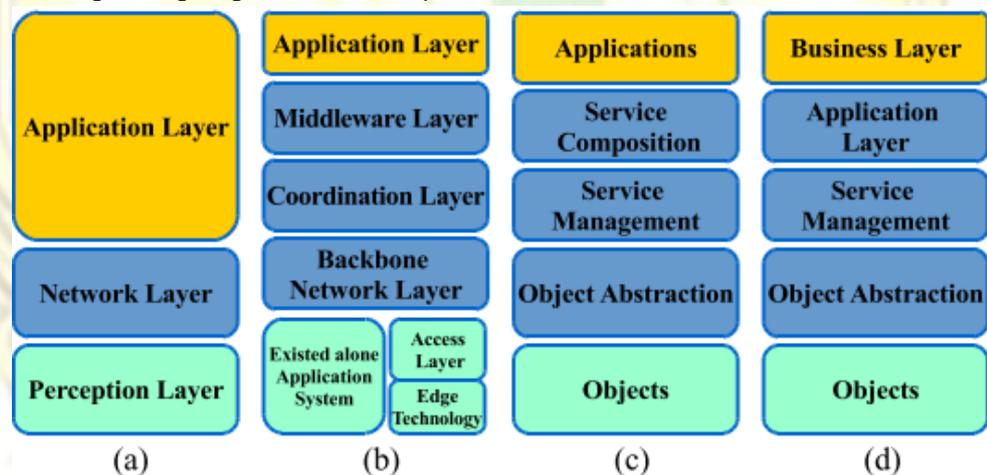
1. Editor program, yang digunakan untuk membuat dan mengedit program dalam bahasa pemrosesan. Program-program pada Arduino disebut sebagai *Sketch*.
2. *Compiler*, modul yang mengubah bahasa pemrosesan (kode program) menjadi kode biner, karena satu-satunya bahasa pemrograman yang dipahami oleh *mikrokontroler* adalah kode biner.
3. *Uploader*, modul yang digunakan untuk memuat kode biner ke dalam memori *mikrokontroler*

### 2.5. Internet of Things (IoT)

*Internet of Things* (IoT) adalah suatu konsep yang melibatkan jaringan terhubung antara perangkat fisik, sensor, dan perangkat lunak. IoT melibatkan semua jenis entitas, termasuk mesin, bangunan, perangkat, hewan, manusia, dan banyak lagi. Dalam pengembangan internet, IoT dianggap sebagai langkah maju. Menurut Komisi Eropa, integrasi internet dengan komunikasi nirkabel dan jaringan sensor nirkabel telah mengubah paradigma dalam transformasi perangkat sehari-hari. IoT mendukung pengembangan perangkat yang cerdas dan responsif terhadap konteks. Dengan struktur teknisnya, pangsa pasar yang luas, nilai yang tinggi, dan pendapatan yang signifikan, IoT telah menemukan tempatnya dalam hampir semua aspek kehidupan manusia. Kehadirannya tidak dapat dihindari[21].

Internet Industri (*Industrial Internet*) menyediakan kemampuan untuk memperoleh visibilitas dan pemahaman tentang operasi dan aset perusahaan dengan mengintegrasikan sensor mesin, *middleware*, perangkat lunak, dan sistem *backend* serta menggunakan komputasi awan untuk penyimpanan data. Internet Industri menggabungkan beberapa teknologi utama untuk menciptakan sistem yang lebih unggul daripada sekadar gabungan komponennya. Salah satu perkembangan teknologi terkini yang signifikan terlihat pada teknologi sensor. Dalam beberapa tahun terakhir, teknologi sensor telah mengalami penurunan biaya dan ukuran yang signifikan, memungkinkan penggunaan yang lebih ekonomis dan teknis pada mesin, proses, dan bahkan perangkat manusia[22].

*Internet of Things* (IoT) adalah jaringan yang terdiri dari perangkat fisik, sensor, kendaraan, aplikasi, dan protokol komunikasi yang saling terhubung untuk mengumpulkan, mengelola, menganalisis, serta bertukar data. Arsitektur IoT menjelaskan struktur sistem yang mencakup elemen fisik, konfigurasi fungsi, prinsip operasional, dan format data yang digunakan. Meskipun belum ada satu arsitektur yang disepakati secara global karena luasnya cakupan IoT, secara umum arsitektur ini terdiri dari beberapa lapisan, yaitu lapisan sensor, komunikasi, manajemen, *cloud*, layanan, dan aplikasi. Berbagai model arsitektur telah diajukan dalam penelitian sebelumnya. Gambar 2.8 memperlihatkan arsitektur umum IoT dengan susunan lapisan, sedangkan Gambar 2.9 menampilkan arsitektur generik untuk penerapan pada *Smart City*.



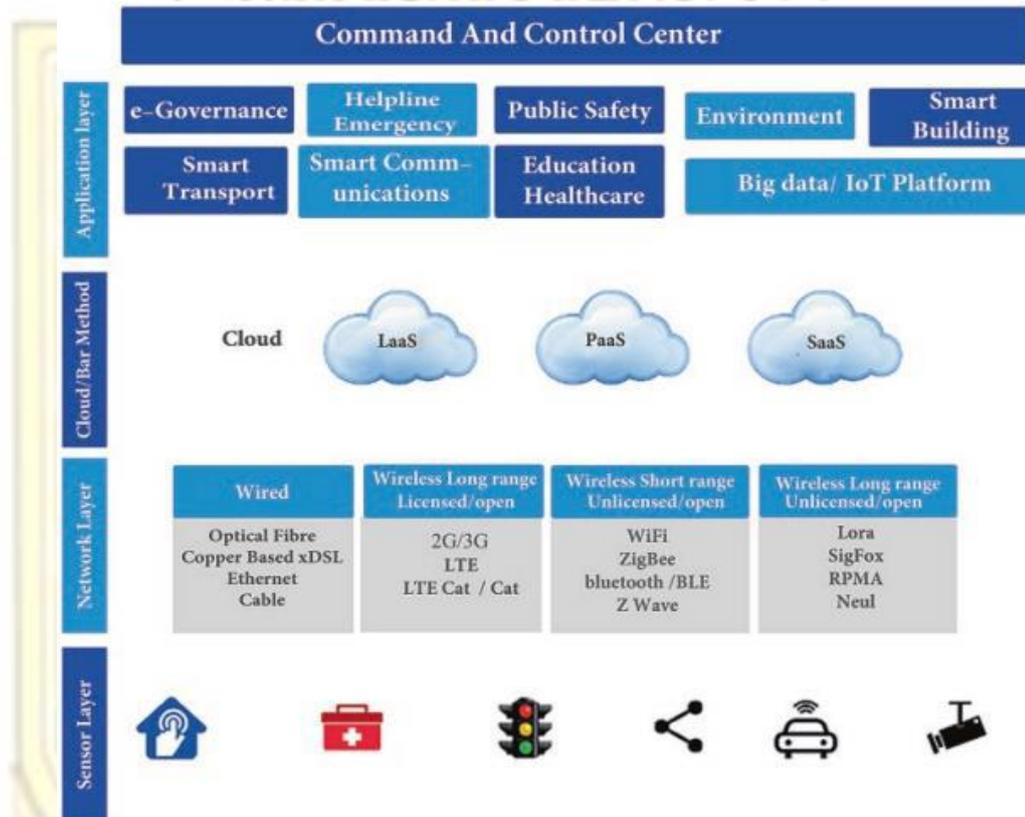
**Gambar 2. 7** Arsitektur Umum IoT[22]

Ada beberapa arsitektur umum yang digunakan dalam IoT:

- a) Arsitektur 3-layer: Terdiri dari lapisan perangkat, jaringan, dan aplikasi.
- b) Arsitektur *Middleware*: Melibatkan lapisan middleware sebagai penghubung antara perangkat keras dan perangkat lunak.

- c) Arsitektur SOA (*Service-Oriented Architecture*): Fokus pada penyediaan layanan yang dapat digunakan oleh aplikasi dalam sistem IoT.
- d) Arsitektur 5-layer: Terdiri dari lapisan perangkat keras, perangkat lunak, jaringan, aplikasi, dan pengguna.

Pilihan arsitektur tergantung pada kebutuhan dan tujuan sistem IoT yang akan diimplementasikan. Gambar 2.9 menggambarkan arsitektur IoT secara umum untuk *Smart City*.



**Gambar 2. 8** Gambaran umum *Smart City* IoT Arsitektur[23]

Menurut mayoritas peneliti, arsitektur *Internet of Things* (IoT) terdiri dari tiga lapisan[23], yaitu:

1. Lapisan Persepsi

Lapisan ini merupakan lapisan fisik yang dilengkapi dengan sensor untuk mengumpulkan informasi dari lingkungan sekitar. Pada lapisan ini, beberapa parameter fisik ditangkap atau objek pintar lainnya diidentifikasi.

## 2. Lapisan Jaringan

Lapisan ini terhubung dengan perangkat cerdas, perangkat jaringan, dan server lainnya. Fungsinya adalah mengirim dan memproses data yang dikumpulkan dari sensor pada lapisan persepsi. Lapisan jaringan bertanggung jawab atas komunikasi dan pengolahan data dalam sistem IoT.

## 3. Lapisan Aplikasi

Lapisan ini menyediakan layanan aplikasi khusus pengguna. Berbagai aplikasi IoT seperti rumah pintar, kota pintar, dan kesehatan cerdas berada pada lapisan ini. Lapisan aplikasi bertanggung jawab atas pengembangan dan penyediaan layanan yang sesuai dengan kebutuhan pengguna dalam konteks IoT.

Dengan adanya tiga lapisan ini, arsitektur IoT memungkinkan pengumpulan data, komunikasi, dan pengembangan aplikasi yang lebih efektif dan terintegrasi dalam ekosistem IoT.

### 2.5.1. *Message Queuing Telemetry Transport (MQTT)*

MQTT adalah protokol komunikasi yang sederhana dan ringan yang dirancang khusus untuk digunakan pada perangkat dengan sumber daya terbatas, *bandwidth* rendah, latensi tinggi, atau jaringan yang tidak stabil. Prinsip desain MQTT adalah mengoptimalkan penggunaan *bandwidth* jaringan dan sumber daya perangkat keras, sambil tetap memastikan tingkat keandalan dan transmisi data yang memadai. MQTT memiliki dua fungsi utama, yaitu publikasi (*publish*) dan berlangganan (*subscribe*) yang dapat dilihat pada Gambar 2.10. Fungsi publikasi digunakan oleh perangkat untuk mengirimkan data atau perintah kepada perangkat lain. Sedangkan fungsi berlangganan digunakan untuk menerima informasi atau perintah yang dikirim oleh perangkat lain yang ingin menerimanya. Dengan menggunakan MQTT, perangkat dapat berkomunikasi secara efisien dan efektif dalam jaringan dengan keterbatasan sumber daya atau kondisi jaringan yang tidak stabil[24].



Gambar 2. 9 Pub/Sub MQTT[24]

### 2.5.2. Node-RED

Node-RED adalah sebuah aplikasi yang dirancang dengan antarmuka visual dan berbasis browser. Aplikasi ini sering digunakan dalam konteks IoT karena kemampuannya dalam mempermudah pengguna dalam membuat aplikasi dengan menggunakan konsep "flow". Dalam Node-RED, pengguna dapat menggabungkan blok-blok kode yang telah ditentukan sebelumnya, yang disebut sebagai "node", untuk menyelesaikan tugas yang diinginkan. Node-RED menggunakan pendekatan pemrograman visual di mana pengembang dapat menghubungkan *node-node* tersebut untuk membentuk "flow" yang mengatur alur kerja aplikasi. *Node-node* yang terhubung dapat berupa *node input*, *node* pengujian, atau *node output*[25]. Gambar 2.8 menunjukkan tampilan antarmuka Node-RED yang menampilkan sebuah *flow* yang akan dibuat.

### 2.5.3. Database

*Database* adalah gudang atau markas tempat berkumpul dan bersarangnya data. Data merupakan representasi fakta-fakta dunia nyata yang mewakili objek seperti manusia, benda, kejadian, dan sebagainya. Data dapat berupa teks, angka, gambar, bunyi, simbol, atau kombinasinya. *Database* adalah kumpulan item data yang saling berhubungan dan diorganisasi berdasarkan skema atau struktur tertentu. *Database* disimpan di dalam *hardware* komputer dan menggunakan *software* untuk melakukan manipulasi data, seperti pembaruan, pencarian, pengolahan dengan perhitungan tertentu, dan penghapusan, dengan tujuan tertentu[26].

### 2.5.4. MySQL

MySQL adalah *Relational Database Management System* (RDBMS) yang gratis dan didistribusikan dengan *General Public Licence* (GPL). Ini adalah sistem manajemen basis data relasional yang populer dengan fitur kuat dan dukungan *Structured Query Language* (SQL). MySQL digunakan secara luas untuk menyimpan, mengelola, dan mengakses data dengan efisiensi tinggi[27]. MySQL adalah server *database* yang sangat populer. Dikenal dengan penggunaan bahasa SQL sebagai bahasa dasar untuk mengakses dan mengelola *database*. MySQL telah menjadi salah satu pilihan utama untuk aplikasi web dan bisnis yang membutuhkan penyimpanan dan pengelolaan data yang efisien[28].

SQL adalah bahasa pemrograman yang digunakan untuk mengelola dan memanipulasi data dalam basis data. MySQL merupakan server *database* yang populer dan mendukung penggunaan SQL. MySQL diklaim memiliki kecepatan *query* yang lebih baik daripada beberapa server *database* lainnya, seperti PostgreSQL dan Interbase. Namun, performa server *database* dapat dipengaruhi oleh faktor-faktor lain. Pemilihan server *database* harus mempertimbangkan kebutuhan proyek atau aplikasi yang spesifik[29].

### 2.5.5. Python

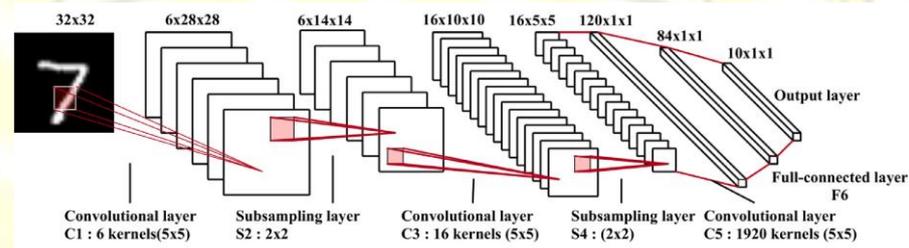
Python adalah bahasa pemrograman populer yang digunakan untuk mengembangkan aplikasi web, *mobile*, dan desktop. Diciptakan pada tahun 1990 oleh Guido van Rossum, Python terkenal karena sintaksisnya yang sederhana dan mudah dipahami. Bahasa ini juga memiliki berbagai pustaka dan modul yang luas, yang memudahkan pengembang dalam membangun aplikasi dengan cepat. Python telah menjadi pilihan yang populer di kalangan pengembang karena kemampuannya dalam mengatasi berbagai tugas pemrograman dengan efisiensi tinggi[30].

### 2.6. Forecasting

Peramalan (*Forecasting*) adalah proses untuk memperkirakan kebutuhan di masa mendatang, termasuk dalam hal kuantitas, kualitas, waktu, dan lokasi yang diperlukan untuk memenuhi permintaan barang atau jasa. Peramalan melibatkan penyusunan rencana berdasarkan kapasitas dan kemampuan permintaan/produksi yang telah ada di perusahaan. Dengan melakukan peramalan yang akurat, perusahaan dapat mengoptimalkan persediaan, mengatur produksi, dan merencanakan kegiatan bisnis lainnya dengan lebih efektif[31]. Tujuan peramalan adalah untuk meramalkan keadaan di masa mendatang dengan mengidentifikasi dan mengukur variabel bebas yang berpengaruh terhadap variabel tak bebas yang diamati. Hal ini dilakukan untuk memenuhi kebutuhan perencanaan jangka panjang dan untuk menghasilkan ramalan yang memiliki tingkat kesalahan yang minimal, yang dapat diukur menggunakan *Mean Absolute Percent Error* (MAPE) atau metode lainnya. Dengan demikian, peramalan bertujuan untuk memberikan informasi yang akurat dan berguna dalam pengambilan keputusan bisnis[32].

### 2.6.1. Convolutional Neural Network (CNN)

CNN (*Convolutional Neural Network*) adalah jenis model *neural network* yang digunakan untuk pengolahan data berstruktur *grid* seperti citra dan teks, dapat dilihat pada Gambar 2.11. CNN memiliki kemampuan untuk menangkap informasi kontekstual dalam data dan memiliki keunggulan seperti sederhana, waktu pelatihan cepat, dan membutuhkan jumlah sampel data pelatihan yang lebih sedikit dibandingkan dengan model *neural network* lainnya. CNN sangat populer dalam tugas-tugas seperti pengenalan gambar, klasifikasi, segmentasi, dan deteksi objek[33].



**Gambar 2. 10** Arsitektur LeNet-5 (CNN oleh LeCun)[33]

Model LeNet5 yang diusulkan oleh LeCun merupakan salah satu model CNN pertama yang menggunakan operasi *convolution* dan *pooling* untuk memanfaatkan informasi struktural dalam data citra. Model ini telah berhasil dalam tugas klasifikasi data citra dan menjadi landasan untuk pengembangan model *deep learning* yang lebih kompleks dalam analisis data citra.

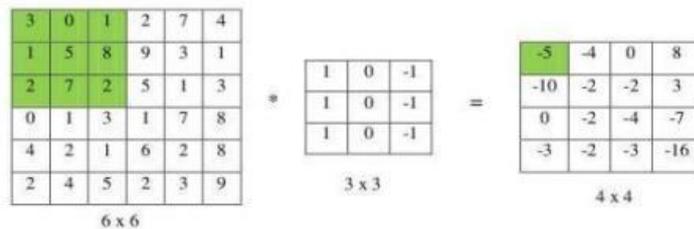
Model *Convolutional Neural Network* (CNN) umumnya terdiri dari beberapa layer, antara lain:

1. *Layer input*: Menerima data *input*.
2. *Layer convolutional*: Melakukan operasi *convolution* pada *input* menggunakan filter untuk mengekstraksi fitur-fitur relevan.
3. *Layer ReLU*: Menerapkan fungsi aktivasi ReLU pada *output* sebelumnya.
4. *Layer pooling*: Melakukan *downsampling* pada *output* sebelumnya untuk mengurangi dimensi data.
5. *Layer fully connected*: Terdiri dari neuron-neuron yang terhubung sepenuhnya, menghitung *output* akhir dari model.

Setiap layer memiliki fungsi dan tugasnya sendiri dalam memproses data dan mempelajari fitur-fitur penting.

### 2.7.1.1. Convolution Layer

Lapisan konvolusi merupakan tahap awal dalam arsitektur *Convolutional Neural Network* (CNN). Pada tahap ini, dilakukan proses konvolusi antara gambar masukan dan filter tertentu, tanpa mengubah struktur dasar dari gambar tersebut. Tujuan utama dari proses ini adalah untuk mengambil atau mengenali ciri-ciri penting dalam gambar. Hasil dari konvolusi ini adalah representasi linear dari data masukan berdasarkan pola spasial yang ada. Bobot yang dimiliki pada lapisan ini membentuk filter konvolusi, dan nilai-nilai bobot tersebut akan diperbarui selama proses pelatihan CNN untuk menyesuaikan dengan data *input* yang digunakan. Contoh visualisasi dari proses ini dapat dilihat pada Gambar 2.12.



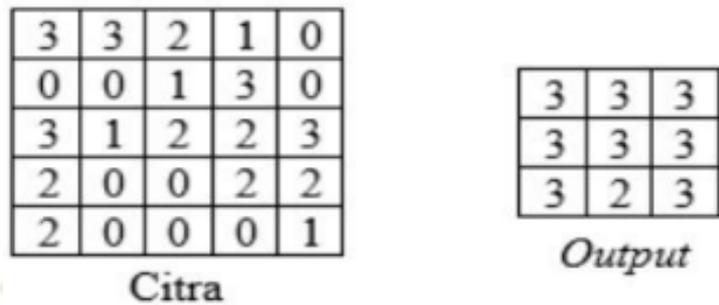
Gambar 2. 11 Convolution Layer[33]

### 2.7.1.2. Pooling Layer

Lapisan *pooling* adalah proses yang digunakan untuk mengurangi jumlah parameter dan perhitungan dalam jaringan, serta mencegah terjadinya *overfitting* pada citra. Lapisan ini terdiri dari dua jenis, yaitu *average pooling* dan *max pooling*. Pada *average pooling*, nilai rata-rata diambil dari area yang dipilih, sedangkan pada *max pooling*, nilai terbesar diambil dari area yang dipilih. Persamaan untuk *max pooling* ditunjukkan dalam Persamaan 2.2.

$$a_j = \text{Max}_{N \times N} a_i^{n \times n} \mu(n, n) \tag{2.2}[33]$$

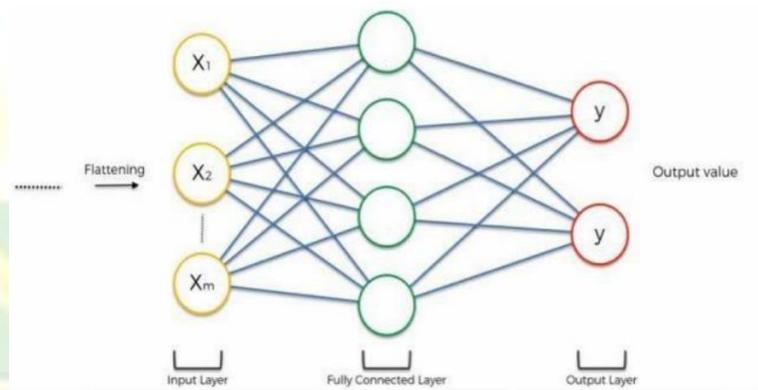
Dalam persamaan tersebut,  $a_i$  mewakili nilai dalam *pooling map*,  $a_j$  mewakili nilai dalam *input map*, dan  $\mu(n, n)$  merupakan fungsi jendela (*window function*) yang digunakan. Contoh dari lapisan *max pooling* ditunjukkan pada Gambar 2.13.



**Gambar 2. 12 Max Pooling Layer**[33]

**2.7.1.3. Fully Connected Layer**

Lapisan *fully connected* adalah bagian dari jaringan yang mengubah semua data menjadi bentuk satu baris (disebut *flatten*), lalu setiap titik datanya terhubung ke semua titik di lapisan berikutnya. Di sini, setiap koneksi punya bobot dan memakai fungsi pengubah sinyal untuk memproses datanya. Hasil akhirnya adalah prediksi yang dibuat berdasarkan data yang masuk. Gambar 2.14 menunjukkan contoh dari lapisan ini.



**Gambar 2. 13 Fully Connected Layer**[33]

**2.7.1.4. Rectified Linear Units (ReLU)**

*Rectified Linear Unit* (ReLU) adalah fungsi pemroses sinyal yang memiliki keunggulan dalam menangani data berukuran besar dengan cepat. Fungsi ini biasanya digunakan di antara lapisan konvolusi dan lapisan *pooling* dalam CNN. ReLU menjaga hasil konvolusi tetap berada pada domain positif, dengan cara mengubah semua nilai negatif menjadi 0. Persamaan matematis dari ReLU ditampilkan pada Persamaan 2.4.

$$f = \max(0, x) \quad (2.4)[33]$$

### 2.6.2. Perhitungan Akurasi *Forecasting*

Dalam mengukur akurasi *forecasting*, terdapat beberapa metode yang umum digunakan seperti *Mean Square Error* (MSE), *Mean Absolute Percentage Error* (MAPE), dan *Mean Percentage Error* (MPE)[34].

#### 2.7.2.1. *Mean Square Error* (MSE):

MSE dihitung dengan mengkuadratkan selisih absolut antara nilai peramalan dan nilai aktual (pengamatan), kemudian membaginya dengan jumlah pengamatan. Formulasnya adalah:

$$MSE = \frac{\sum_{t=1}^n (y_t - \tilde{y}_t)^2}{n} \quad (2.5)[34]$$

Keterangan :

$y_t$  : Realisasi pada periode t

$\tilde{y}_t$  : Ramalan pada periode t

n : Banyaknya observasi

t : 1,2,3,.....

#### 2.7.2.2. *Mean Absolute Percentage Error* (MAPE):

MAPE mengukur persentase kesalahan absolut rata-rata dalam peramalan dibandingkan dengan nilai aktual. Formulasnya adalah:

$$MAPE = \frac{\sum_{t=1}^n \frac{|y_t - \tilde{y}_t|}{y_t}}{n} \times 100\% \quad (2.6)[34]$$

Keterangan :

$y_t$  : Realisasi pada periode t

$\tilde{y}_t$  : Ramalan pada periode t

n : Banyaknya observasi

t : 1,2,3,.....

#### 2.7.2.3. *Mean Percentage Error* (MPE):

MPE mengukur rata-rata persentase kesalahan dalam peramalan dibandingkan dengan nilai aktual. Formulasnya adalah:

$$MPE = \frac{\sum_{t=1}^n \frac{y_t - \tilde{y}_t}{y_t}}{n} \times 100\%$$

(2.7)[34]

Keterangan :

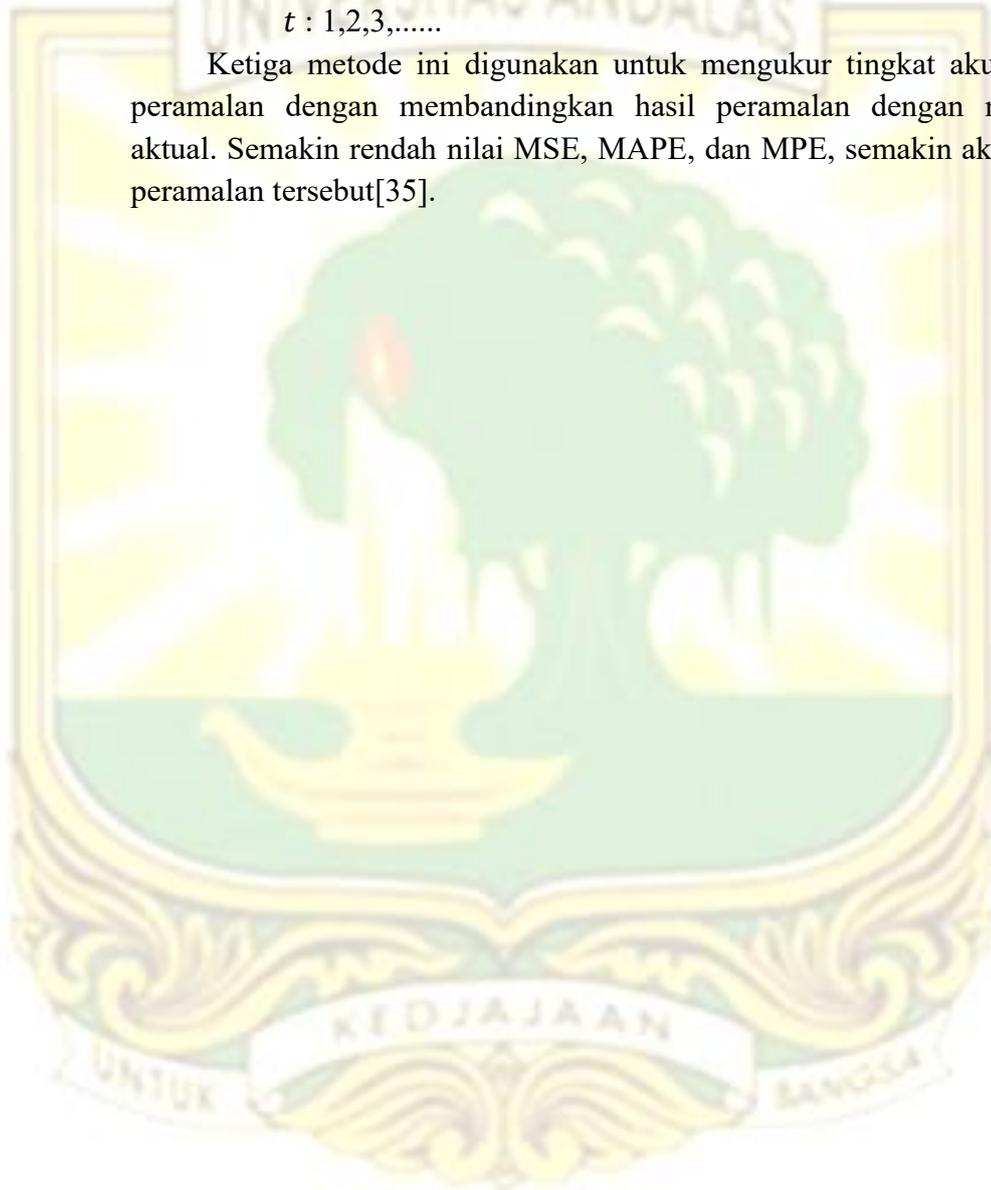
$y_t$  : Realisasi pada periode t

$\tilde{y}_t$  : Ramalan pada periode t

$n$  : Banyaknya observasi

$t$  : 1,2,3,.....

Ketiga metode ini digunakan untuk mengukur tingkat akurasi peramalan dengan membandingkan hasil peramalan dengan nilai aktual. Semakin rendah nilai MSE, MAPE, dan MPE, semakin akurat peramalan tersebut[35].



## BAB III METODE PENELITIAN

### 3.1. Jenis Penelitian

Dalam tugas akhir ini, digunakan metode penelitian kuantitatif yang didasarkan pada pengamatan permasalahan yang dihadapi. Pendekatan kuantitatif menekankan pada adanya variabel-variabel sebagai objek penelitian, dan variabel-variabel tersebut didefinisikan dalam bentuk kinerja sistem atau variabel yang mempengaruhi kinerja dan performa sistem[36].

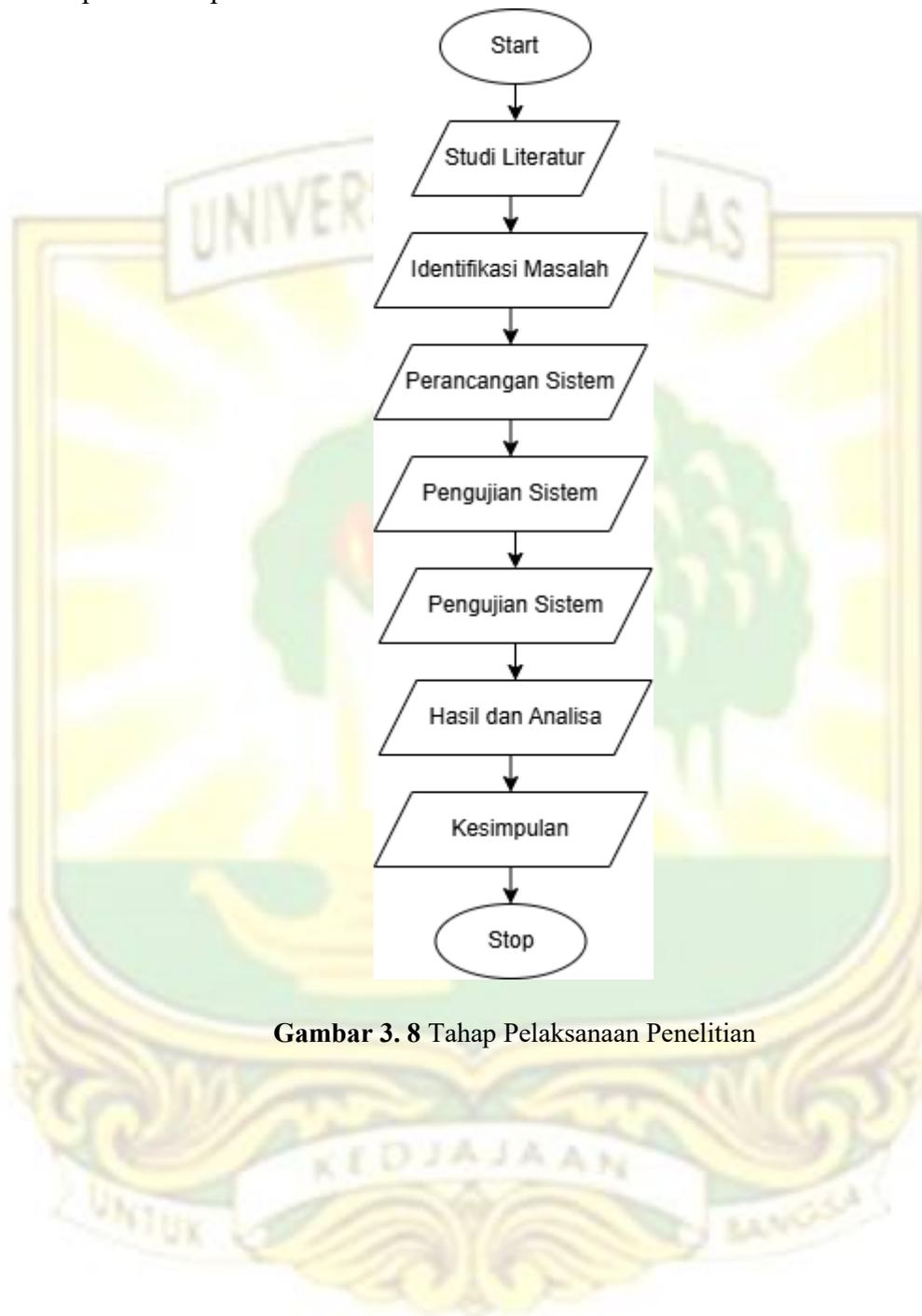
Penelitian ini menggunakan pendekatan eksperimental dengan mengobservasi sistem yang akan dirancang, sehingga dapat menghasilkan data berdasarkan kondisi lapangan. Sistem yang dirancang mengumpulkan data aktual sesuai dengan kondisi yang ada di lapangan. Setelah memperoleh hasil dari sistem yang dirancang, data tersebut dikumpulkan dalam *database* yang digunakan untuk proses analisis dan peramalan.

### 3.2. Tempat dan Waktu Pelaksanaan

Tempat penelitian akan dilakukan di Pondok Pesantren Harkatul Quran, Batang Kabung Ganting, Koto Tangah, Padang, Sumatera Barat. Waktu penelitian dilakukan mulai dari bulan Januari 2024 hingga Juni 2025.

### 3.3. Tahap Pelaksanaan Penelitian

Dalam penelitian ini, penulis menggunakan tahapan penelitian yang dapat dilihat pada Gambar 3.1.



**Gambar 3. 8** Tahap Pelaksanaan Penelitian

Berikut penjelasan tahapan pelaksanaan penelitian yang akan dilakukan pada Gambar 3.1, dapat dijelaskan sebagai berikut.

1. Studi Literatur

Dalam penelitian ini, dilakukan studi literatur untuk mengumpulkan referensi dan teori yang mendukung pembuatan tugas akhir. Studi literatur mencakup topik-topik seperti Genset, akuisisi data, pemrosesan data, dan transmisi data pada ESP32, serta pembuatan antarmuka *monitoring* dengan Node-Red. Referensi yang digunakan meliputi buku, jurnal, dan *datasheet* yang relevan dengan penelitian ini. Penulis juga mempelajari secara langsung penerapan IoT, *machine learning*, penggunaan MQTT broker, serta proses penyimpanan data ke dalam *database*. Selain itu, penulis juga melakukan proses peramalan menggunakan metode jaringan saraf tiruan, khususnya CNN.

1. Identifikasi Masalah

Identifikasi masalah terkait penggunaan genset, data penggunaan, dan kerusakan genset memunculkan kebutuhan akan sistem *monitoring* dan peramalan tingkat kerusakan genset. Sistem ini bertujuan meningkatkan kinerja genset melalui pemantauan *real-time* dan prediksi kerusakan berdasarkan data masa lalu dan sekarang. Dengan implementasi sistem ini, diharapkan dapat meningkatkan efisiensi operasional, mengurangi risiko kerusakan, dan memastikan pemeliharaan tepat waktu.

2. Perancangan Sistem

Penelitian ini melibatkan perancangan sistem yang terbagi menjadi dua bagian utama, yaitu perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras mencakup rangkaian sistem *monitoring* genset yang terdiri dari beberapa sensor, seperti PZEM-004T untuk sensor daya, dan MPU6050 untuk sensor getaran dan suhu. Sistem ini dikendalikan oleh ESP32 dan terhubung dengan protokol MQTT broker melalui jaringan Wi-Fi, sehingga memungkinkan pemantauan secara daring. Sementara itu, perancangan perangkat lunak melibatkan proses penyimpanan data ke dalam *database* dan melibatkan analisis data, termasuk proses peramalan. Dalam penelitian ini, peramalan dilakukan menggunakan algoritma jaringan saraf tiruan.

### 3. Pengujian Sistem

Pada tahap ini, dilakukan pengujian kinerja sistem *monitoring* dan pengambilan data untuk proses peramalan serta analisis. Pengujian sistem *monitoring* melibatkan verifikasi konsistensi data yang dikirim dan diterima, sementara pengujian pada komputasi peramalan melibatkan evaluasi akurasi dan tingkat kesalahan dari hasil peramalan. Proses pengambilan data dilakukan di Pondok Pesantren Harkatul Quran yang terletak di Batang Kabung Ganting, Koto Tangah, Padang, Sumatera Barat.

### 4. Hasil dan Analisa

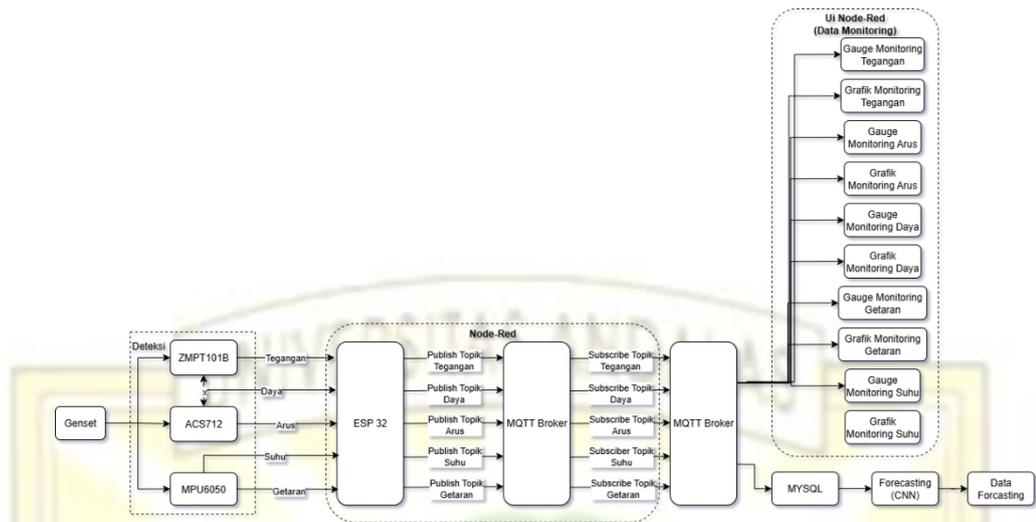
Data hasil pengujian perancangan digunakan untuk menyimpulkan kinerja sistem *monitoring* dan peramalan pada genset secara keseluruhan. Kesimpulan tersebut digunakan sebagai dasar untuk melakukan perbaikan pada sistem guna mengatasi kekurangan yang teridentifikasi. Dengan melakukan perbaikan, diharapkan kinerja sistem dapat meningkat.

### 5. Kesimpulan

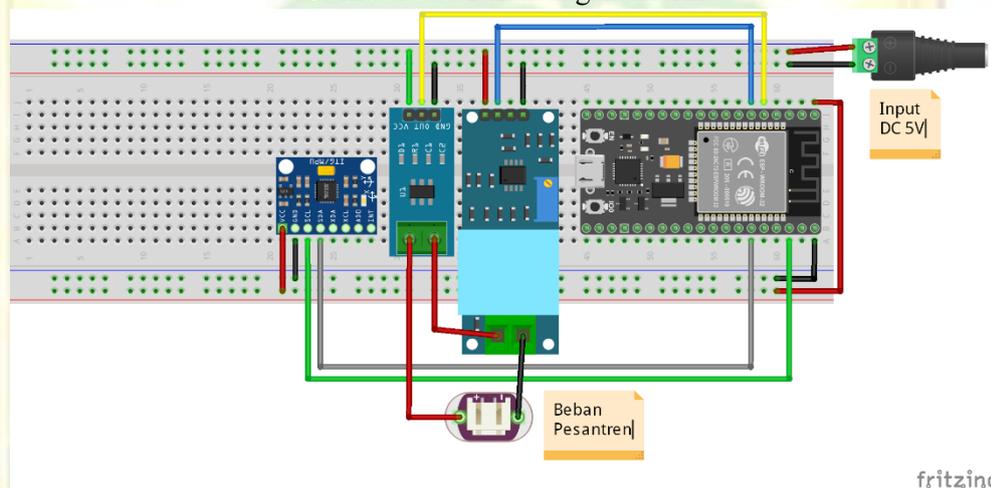
Kesimpulan diperoleh dari keseluruhan hasil yang diperoleh pada penelitian yang telah dilakukan.

## 3.4. Perancangan Sistem

Tugas akhir ini membagi proses perancangan sistem menjadi dua komponen utama, sebagaimana ditunjukkan pada Gambar 3.9, yaitu perancangan perangkat keras dan perangkat lunak. Sementara itu, Gambar 3.10 menggambarkan rangkaian alat yang digunakan. Perangkat keras terdiri atas modul ZMPT101B, ACS712, MPU6050, dan ESP32 DEVKIT V1 yang berperan dalam memantau daya keluaran serta parameter fisik pada genset. Sensor-sensor tersebut bertugas mengukur tegangan, arus, getaran, dan suhu dari genset, lalu mengirimkan data tersebut ke Node-Red melalui koneksi Wi-Fi dengan menggunakan protokol MQTT. Data yang diterima kemudian ditampilkan secara *real-time* pada *dashboard* Node-Red yang telah dirancang sebelumnya.



**Gambar 3. 9** Perancangan Sistem

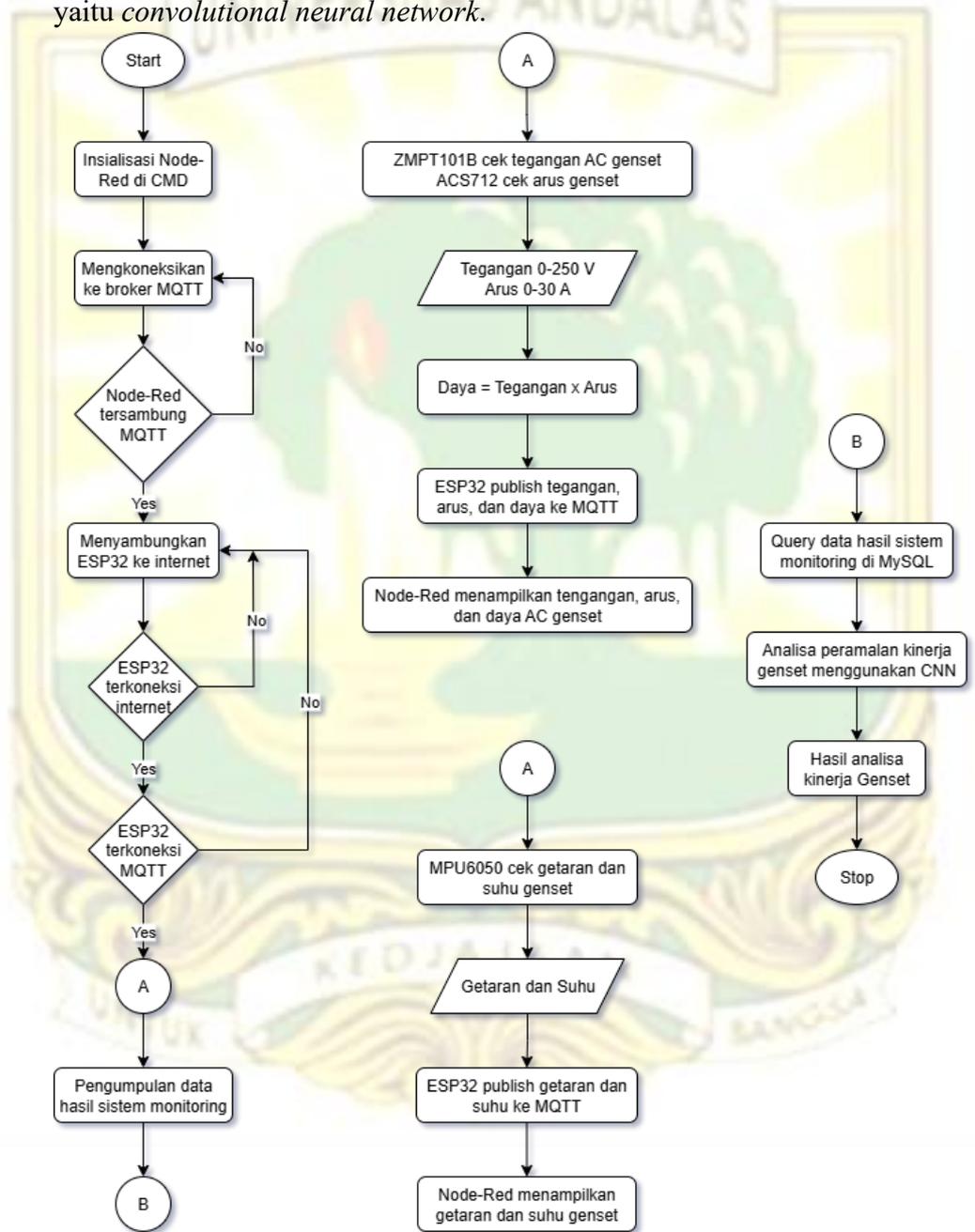


**Gambar 3. 10** Rangkaian Alat

Alur kerja pengiriman data *real-time* hasil sistem *monitoring* genset dapat dilihat pada Gambar 3.9. Baik ESP32 maupun Node-RED diawali dengan proses penyambungan keduanya ke jaringan internet. Sensor-sensor analog yang digunakan akan mengirimkan data sesuai dengan topik masing-masing sensor ke Node-RED. Sensor ZMPT101B digunakan untuk mendeteksi tegangan AC yang disambungkan secara paralel dengan beban, sedangkan ACS712 digunakan untuk mengukur arus dan dipasang secara seri dengan beban. Pembacaan nilai tegangan dan arus dilakukan melalui konversi nilai analog (ADC) dari masing-masing sensor yang telah dikalibrasi. Sensor MPU6050 digunakan untuk mendeteksi perubahan posisi dalam sumbu x, y, dan z, yang kemudian dianalisis untuk mendapatkan nilai percepatan guna menentukan apakah getaran genset masih dalam kategori aman. Selain itu, MPU6050 juga memiliki sensor suhu internal yang dapat digunakan untuk memantau temperatur di sekitar sensor.

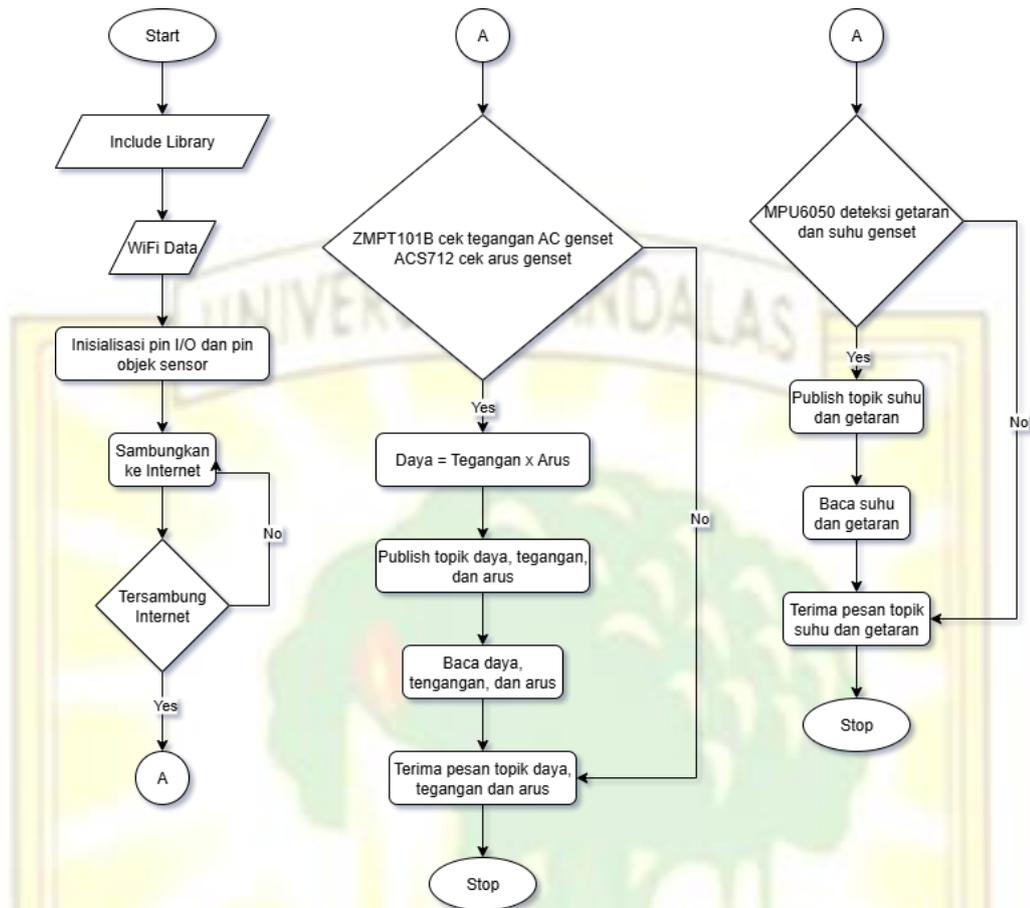
Kemudian ESP32 mengirimkan data hasil pengukuran sensor menggunakan sensor yang ditampilkan pada *dashboard* NodeRED. Alur sistem Pengumpulan data hasil *monitoring* dikirimkan ke *database* Mysql dapat dilihat pada Gambar 3.11.

Setelah pengumpulan data sistem *monitoring* dari *database* Mysql kemudian dilakukan permintaan (*query*) data dari hasil sistem *monitoring* di Mysql untuk melakukan implementasi penggunaan *database* berupa proses analisa peramalan kinerja genset menggunakan konsep *machine learning* yaitu *convolutional neural network*.



Gambar 3. 11 Diagram Alir Sistem

### 3.5. Perancangan *Monitoring* dengan ESP32



**Gambar 3. 12** Diagram Alir Program ESP32

Gambar 3.12 menunjukkan *flowchart* dari program awal yang ditulis menggunakan perangkat lunak Arduino IDE untuk *mikrokontroler* ESP32. Program ini dirinci sebagai berikut:

1. Inisiasi *Library* dan Koneksi Wi-Fi

Program dimulai dengan memasukkan data *library* yang diperlukan untuk menjalankan program. Selain itu, program juga mencakup pengaturan koneksi Wi-Fi, seperti pengisian *username*, *password*, dan *port* MQTT yang akan digunakan.

2. *Inisialisasi* Pin ESP32

Selanjutnya, program melakukan *inisialisasi pin* pada *mikrokontroler* ESP32 yang akan digunakan untuk menghubungkan dengan sensor ACS712, ZMPT101B, dan MPU6050.

3. Koneksi Internet

Setelah meng-*inisialisasi pin*, program berusaha menghubungkan ESP32 ke internet melalui koneksi Wi-Fi yang telah dikonfigurasi sebelumnya. Ini diperlukan agar ESP32 dapat berkomunikasi dengan broker MQTT dan mentransmisikan data sensor ke Node-RED.

4. Koneksi ke Broker MQTT

Ketika ESP32 telah terhubung ke internet, program akan otomatis terhubung ke broker MQTT. Ini memungkinkan ESP32 untuk mengirim data dari sensor PZEM-004T dan MPU6050 ke Node-RED melalui protokol MQTT.

5. *Publish* Data Sensor PZEM-004T

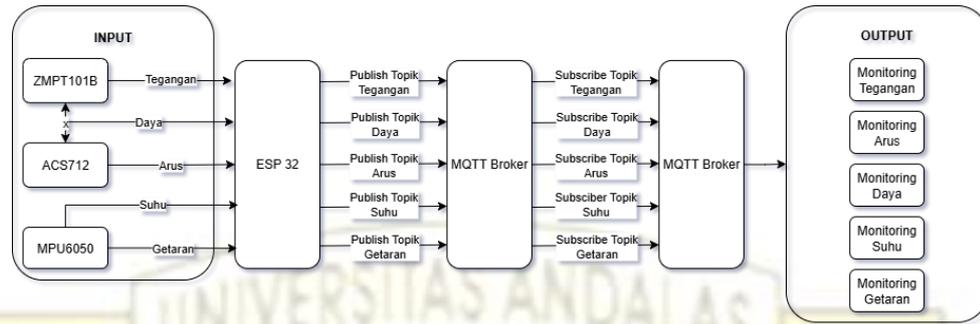
Program akan mem-*publish* data yang diterima dari sensor PZEM-004T ke broker MQTT pada topik yang sesuai, seperti daya, tegangan, dan arus. Data ini akan di-*subscribe* oleh Node-RED untuk pemrosesan lebih lanjut.

6. *Publish* Data Sensor MPU6050

Program juga akan mem-*publish* data yang diterima dari sensor MPU6050 ke broker MQTT pada topik yang sesuai, seperti getaran dan suhu. Node-RED akan melakukan *subscribe* pada topik ini untuk memantau dan memproses data tersebut.

Dengan demikian, program ini memungkinkan ESP32 untuk mengambil data dari sensor PZEM-004T dan MPU6050, mengirimkannya ke Node-RED melalui broker MQTT, dan memungkinkan Node-RED untuk melakukan pemrosesan lebih lanjut berdasarkan data yang diterima. Ini adalah langkah awal dalam menghubungkan ESP32 dengan sensor dan mengintegrasikannya ke dalam sistem IoT yang lebih besar.

### 3.6. Perancangan MQTT



Gambar 3. 13 Diagram Publish dan Subscribe

Gambar 3.13 menjelaskan topik yang dipublikasikan ke broker MQTT, yaitu Mosquitto. ESP32 akan mempublikasikan topik-topik yang berasal dari *input* PZEM-004T dan MPU6050. Topik-topik ini akan di-*subscribe* oleh NodeRED dan akan ditampilkan pada antarmuka *dashboard* NodeRED. *Dashboard* ini berisi pemantauan dari PLTS dan juga kondisi meteorologi di sekitar PLTS.

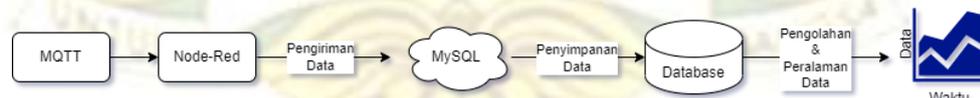
Perlu dicatat bahwa protokol komunikasi yang digunakan dalam MQTT sebenarnya berdasarkan protokol TCP/IP. Jadi, awalnya, koneksi TCP akan dibentuk, dan setelahnya, koneksi MQTT akan dibangun. Setelah koneksi MQTT aktif, transfer data antara perangkat-perangkat terhubung akan terjadi.

TCP	127.0.0.1:1001	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1883	0.0.0.0:0	LISTENING
TCP	127.0.0.1:5000	0.0.0.0:0	LISTENING

Gambar 3. 14 Status Mosquitto MQTT Broker

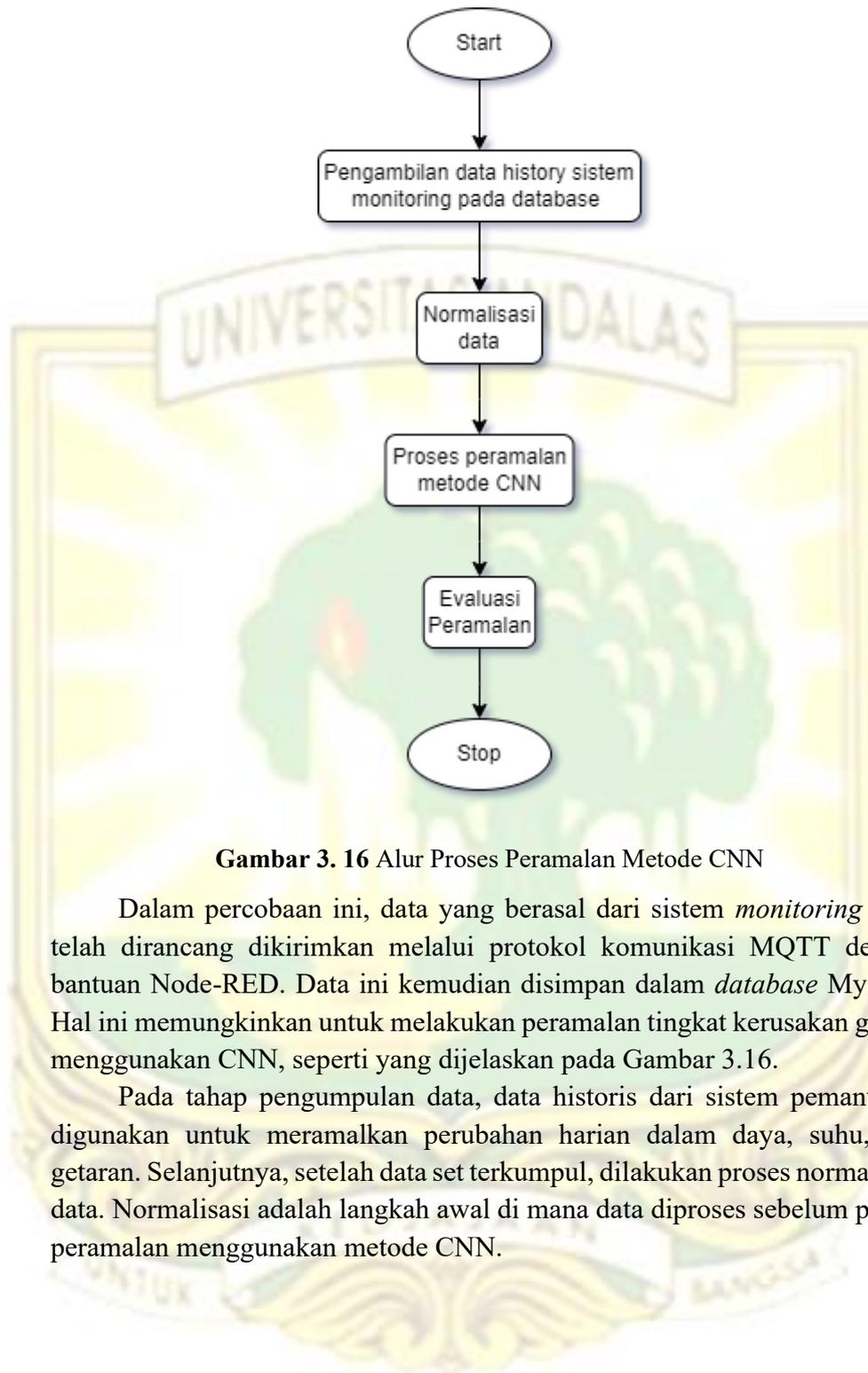
Gambar 3.14 yang Anda tandai dengan lingkaran merah di atas menunjukkan bahwa protokol komunikasi data MQTT broker beroperasi di *port* TCP 1883. Selain itu, ada juga klien dengan alamat NEAN-PAD:0 yang beroperasi melalui protokol TCP/IP. *Default port* yang digunakan dalam protokol MQTT adalah 1883, sedangkan *port* 8883 biasanya digunakan untuk komunikasi aman (*secure*) melalui TLS/SSL.

### 3.7. Peramalan Tingkat Kerusakan



Gambar 3. 15 Diagram Proses Peramalan dari Database

Gambar 3.15 menggambarkan blok diagram dari peramalan tingkat kerusakan genset dalam tugas akhir ini. Sedangkan, alur kerja peramalan secara lebih rinci diilustrasikan pada Gambar 3.16 di bawah ini.



**Gambar 3. 16** Alur Proses Peramalan Metode CNN

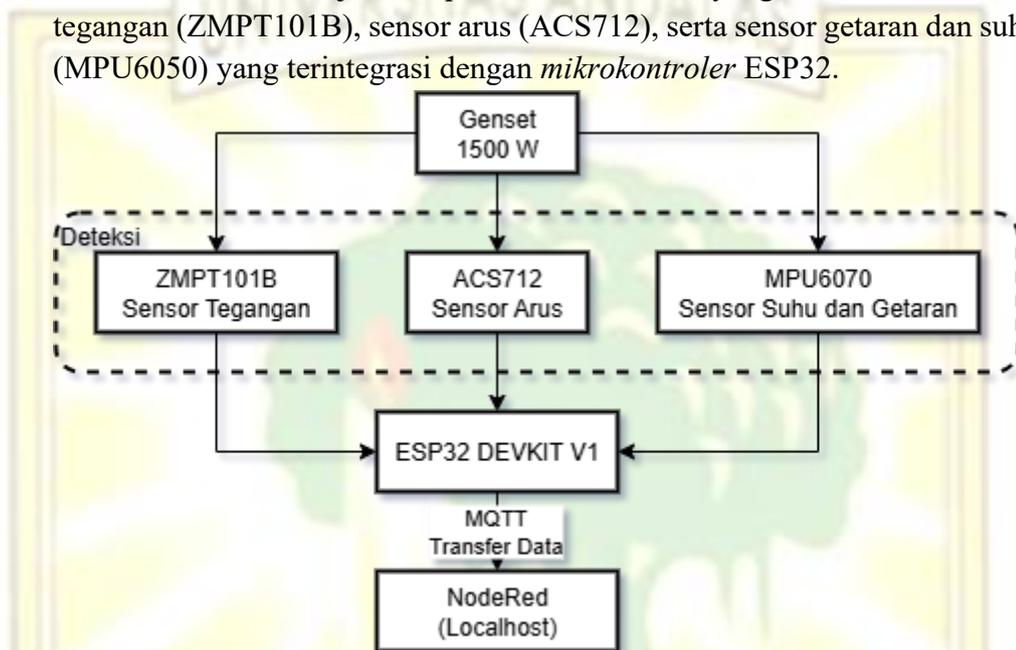
Dalam percobaan ini, data yang berasal dari sistem *monitoring* yang telah dirancang dikirimkan melalui protokol komunikasi MQTT dengan bantuan Node-RED. Data ini kemudian disimpan dalam *database* MySQL. Hal ini memungkinkan untuk melakukan peramalan tingkat kerusakan genset menggunakan CNN, seperti yang dijelaskan pada Gambar 3.16.

Pada tahap pengumpulan data, data historis dari sistem pemantauan digunakan untuk meramalkan perubahan harian dalam daya, suhu, dan getaran. Selanjutnya, setelah data set terkumpul, dilakukan proses normalisasi data. Normalisasi adalah langkah awal di mana data diproses sebelum proses peramalan menggunakan metode CNN.

## BAB IV HASIL DAN ANALISA

### 4.1. Implementasi Desain Sistem *Monitoring*

Diagram hasil perancangan sistem *monitoring* genset ditampilkan pada Gambar 4.1. Setelah seluruh komponen dirancang dan dirakit, hasil fisik dari sistem tersebut ditunjukkan pada Gambar 4.2, yang terdiri atas sensor tegangan (ZMPT101B), sensor arus (ACS712), serta sensor getaran dan suhu (MPU6050) yang terintegrasi dengan *mikrokontroler* ESP32.



**Gambar 4. 1** Diagram Hasil Perancangan Sistem *Monitoring*

Hasil dari perancangan sistem *monitoring* yang ditunjukkan pada Gambar 4.1 dirancang untuk mengirimkan data pembacaan dari sensor tegangan (ZMPT101B), sensor arus (ACS712), serta sensor suhu dan getaran (MPU6050) melalui protokol komunikasi MQTT yang terhubung melalui jaringan Wi-Fi. Data hasil *monitoring* kemudian ditampilkan menggunakan antarmuka Node-RED dan disimpan dalam basis data untuk keperluan pencatatan dan analisis. Tampilan antarmuka sistem pada Node-RED dapat dilihat pada Gambar 4.3, sedangkan Gambar 4.4 menunjukkan hasil pengukuran tegangan dan arus secara langsung.



(a)



(b)

**Gambar 4. 2** Rangkaian sistem *monitoring*: (a) rangkaian terdiri dari ZMPT101B, ACS712, ESP32, dan catu daya 5V; (b) MPU6050 yang diposisikan pada bagian mesin genset.

Hasil pembacaan dari alat yang telah dirancang sebagaimana ditunjukkan pada Gambar 4.2 ditampilkan melalui *dashboard* Node-RED yang terhubung ke *localhost* pada komputer atau laptop. Untuk menguji keakuratan sistem *monitoring* ini, hasil pada tampilan *dashboard* Node-RED (Gambar 4.3) dibandingkan dengan hasil pengukuran manual menggunakan multimeter Junejour seperti yang ditunjukkan pada Gambar 4.4, yang mencakup pembacaan arus dan tegangan secara langsung.



**Gambar 4.3** Antarmuka *Dashboard* NodeRED untuk Menampilkan Sistem *Monitoring*

Dapat dilihat pada Gambar 4.4 hasil pengukuran arus dengan *clamp meter* 2,68 A sama dengan yang ditampilkan pada *dashboard* Node-RED Gambar 4.3 arus yang terukur sebesar 2,68 A dan tegangan sebesar 219 V.

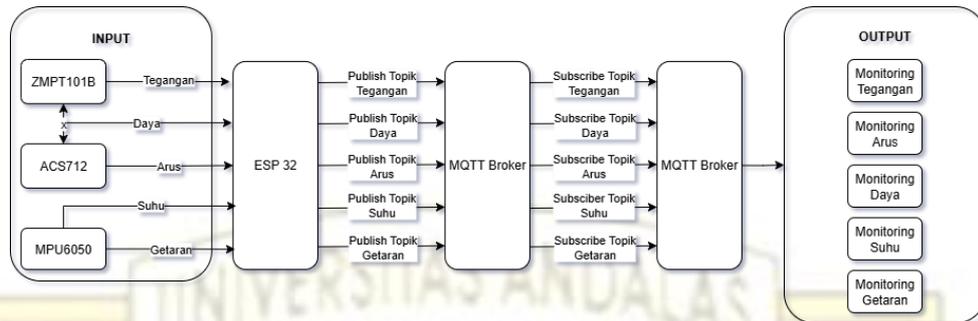


(a)

(b)

**Gambar 4.4** Pengukuran Secara Manual: (a) Arus dengan *Clamp Meter* dan (b) Tegangan dengan Voltmeter

## 4.2. Proses Transmisi Data Hasil Pemantauan Genset Secara *Real-time*



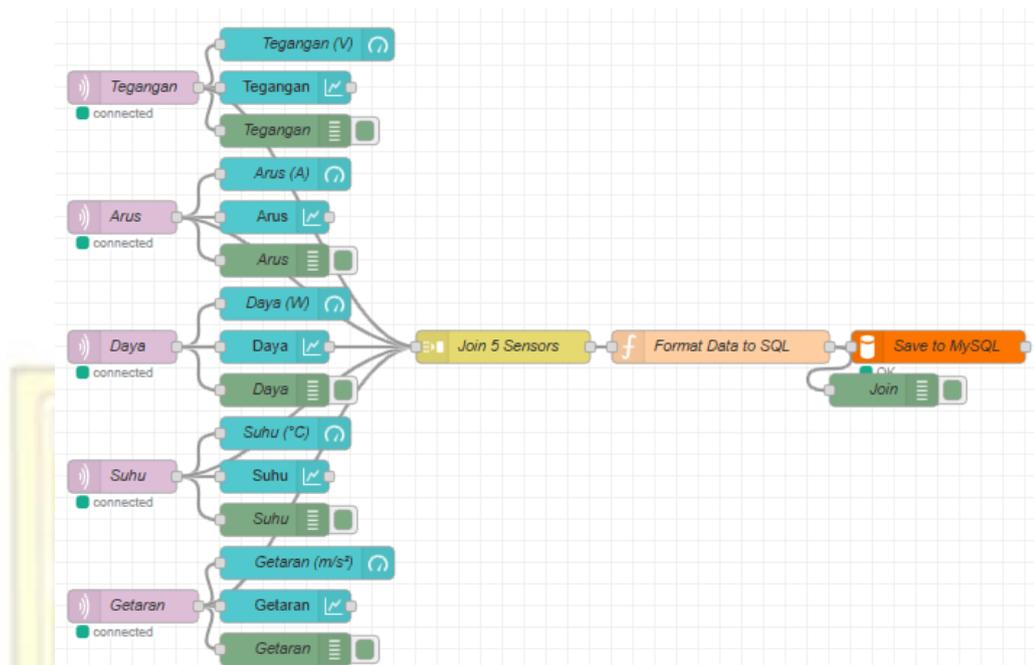
**Gambar 4.5** Diagram *Publish* dan *Subscribe* MQTT

Gambar 4.5 menunjukkan skema topik yang dikirimkan (*publish*) oleh ESP32 ke broker MQTT, yaitu Mosquitto. ESP32 mengirimkan data dari sensor ZMPT101B, ACS712, dan MPU6050 ke masing-masing topik. Topik-topik tersebut kemudian diterima (*subscribe*) oleh Node-RED dan ditampilkan dalam *dashboard* seperti yang terlihat pada Gambar 4.3, yang memvisualisasikan pemantauan daya keluaran serta kondisi genset. MQTT sendiri bekerja di atas protokol TCP/IP, sehingga sebelum data dapat dikirim, terlebih dahulu dilakukan pembentukan koneksi TCP, diikuti oleh *inisialisasi* koneksi MQTT dan proses transfer data.

TCP	127.0.0.1:1001	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1883	0.0.0.0:0	LISTENING
TCP	127.0.0.1:5000	0.0.0.0:0	LISTENING

**Gambar 4.6** Status Mosquitto MQTT Broker

Gambar 4.6 yang diberi penanda lingkaran merah memperlihatkan bahwa protokol komunikasi data melalui MQTT broker berjalan pada *port* TCP 1883. Selain itu, terlihat adanya klien dengan alamat 0.0.0.0:0 yang beroperasi menggunakan protokol TCP/IP, di mana *port default* yang digunakan adalah 1883 dan 8883.



**Gambar 4. 7** Tampilan *Flow Node* untuk Sistem *Monitoring*

Gambar 4.7 di atas menampilkan *Flow Node* berwarna ungu yang berfungsi sebagai *node* untuk *subscribe* topik-topik MQTT pada platform Node-RED. Masing-masing topik mewakili data dari sensor yang terhubung dalam sistem, antara lain: topik Tegangan untuk membaca data dari sensor ZMPT101B, topik Arus dari sensor ACS712, topik Daya yang merupakan hasil perhitungan antara tegangan dan arus, topik Suhu dari sensor suhu MPU6050, serta topik Getaran dari sensor getaran MPU6050.

#### 4.3. Tahapan Penyimpanan Data *Monitoring* ke *Database* MySQL di *Localhost*

Gambar 4.8 menunjukkan tabel berisi data hasil pengukuran dari sistem *monitoring* yang tersimpan dalam *database*. Data ini kemudian diekstrak dan disimpan dalam format CSV, yang dapat diunduh sebagaimana terlihat pada Gambar 4.9. Tampilan *file* CSV ketika dibuka di Excel ditunjukkan pada Gambar 4.10.

	id	tegangan	arus	daya	suhu	getaran	waktu
<input type="checkbox"/>	0	217.6	3.78	821.84	91.48	0.87	2025-08-29 11:00:10
<input type="checkbox"/>	1	217.82	2.12	461.38	91.47	0.88	2025-08-29 11:15:17
<input type="checkbox"/>	2	217.71	2.73	593.79	91.49	0.84	2025-08-29 11:30:08
<input type="checkbox"/>	3	215.98	3.29	710.55	91.47	0.87	2025-08-29 11:45:25
<input type="checkbox"/>	4	215.52	3.57	769.8	91.47	0.74	2025-08-29 12:00:52
<input type="checkbox"/>	5	213.43	2.41	514.49	91.47	0.88	2025-08-29 12:15:47
<input type="checkbox"/>	6	208.81	2.75	573.07	91.48	0.82	2025-08-29 12:30:28
<input type="checkbox"/>	7	208.76	2.42	505.21	91.45	0.83	2025-08-29 12:45:15
<input type="checkbox"/>	8	207.83	2.81	583.14	91.47	0.7	2025-08-29 13:00:20
<input type="checkbox"/>	9	209.32	2.15	450.03	91.47	0.72	2025-08-29 13:15:15
<input type="checkbox"/>	10	212.21	2.52	535.14	91.46	0.87	2025-08-29 13:30:13
<input type="checkbox"/>	11	201.12	2.8	583.02	91.49	0.84	2025-08-29 13:45:13
<input type="checkbox"/>	12	202.86	3	607.88	91.47	0.88	2025-08-29 14:00:33
<input type="checkbox"/>	13	210.23	3.27	688.87	91.47	0.88	2025-08-29 14:15:53
<input type="checkbox"/>	14	208.89	3.88	758.14	91.45	0.83	2025-08-29 14:30:24
<input type="checkbox"/>	15	213.58	3.88	820.91	91.41	0.88	2025-08-29 14:45:28
<input type="checkbox"/>	16	218.5	3	654.89	91.51	0.71	2025-08-29 15:00:39
<input type="checkbox"/>	17	208.19	2.98	608.5	91.83	0.71	2025-08-29 15:15:04
<input type="checkbox"/>	18	214.08	3.1	664.47	91.77	0.88	2025-08-29 15:30:08
<input type="checkbox"/>	19	213.88	3.18	673.23	91.92	0.89	2025-08-29 15:45:51
<input type="checkbox"/>	20	218.78	3.43	744.33	91.88	0.88	2025-08-29 16:00:39
<input type="checkbox"/>	21	218.72	3.85	841.01	91.83	0.85	2025-08-29 16:15:46
<input type="checkbox"/>	22	218.89	3.24	712.79	91.85	0.86	2025-08-29 16:30:32
<input type="checkbox"/>	23	199.93	2.54	507.83	91.92	0.71	2025-08-29 16:45:19

Gambar 4. 8 Tampilan phpMyAdmin di localhost

#### 4.4. Implementasi Database Untuk Peramalan Menggunakan Metode CNN

Gambar 4.8 menampilkan tabel yang berisi kumpulan data hasil pengukuran dari sistem *monitoring*. Data yang telah tersimpan dalam *database* ini selanjutnya berfungsi sebagai gudang data, yang kemudian diekstrak dan disimpan dalam format CSV. *File* CSV tersebut dapat diunduh, seperti yang ditunjukkan pada Gambar 4.9. Dan untuk tampilan data csv di Excel bisa dilihat pada Gambar 4.10.

Exporting rows from "monitoring\_genset" table

Export method:

- Quick - display only the minimal options
- Custom - display all possible options

Format:

SQL

Rows:

- Dump all rows
- Dump some row(s)

Number of rows:

87

Row to begin at:

0

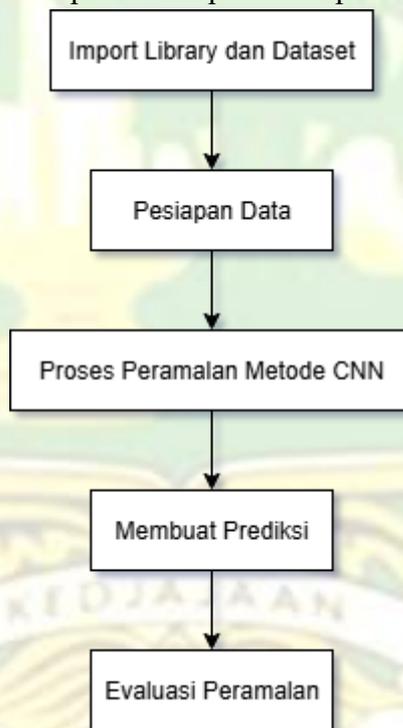
Export

Gambar 4. 9 Antarmuka Proses Ekspor Database MySQL melalui PhpMyAdmin localhost

A1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	id,"tegangan","arus","daya","suhu","getaran","waktu"																			
2	0,"217.6","3.78","821.84","91.48","0.67","2025-06-29 11:00:10"																			
3	1,"217.62","2.12","461.36","91.47","0.66","2025-06-29 11:15:17"																			
4	2,"217.71","2.73","593.79","91.49","0.64","2025-06-29 11:30:05"																			
5	3,"215.98","3.29","710.55","91.47","0.74","2025-06-29 11:45:25"																			
6	4,"215.52","3.57","769.9","91.47","0.74","2025-06-29 12:00:52"																			
7	5,"213.48","2.41","514.49","91.47","0.66","2025-06-29 12:15:47"																			
8	6,"208.61","2.75","573.07","91.48","0.62","2025-06-29 12:30:28"																			
9	7,"208.76","2.42","505.21","91.45","0.63","2025-06-29 12:45:15"																			
10	8,"207.83","2.81","583.14","91.47","0.7","2025-06-29 13:00:20"																			
11	9,"209.32","2.15","450.03","91.47","0.72","2025-06-29 13:15:15"																			
12	10,"212.21","2.52","535.14","91.46","0.67","2025-06-29 13:30:13"																			
13	11,"201.12","2.8","563.02","91.49","0.64","2025-06-29 13:45:18"																			
14	12,"202.66","3","607.98","91.47","0.66","2025-06-29 14:00:33"																			
15	13,"210.23","3.27","686.87","91.47","0.68","2025-06-29 14:15:53"																			
16	14,"208.99","3.68","769.14","91.45","0.63","2025-06-29 14:30:34"																			
17	15,"213.58","3.89","829.91","91.41","0.68","2025-06-29 14:45:26"																			
18	16,"218.5","3","654.99","91.51","0.71","2025-06-29 15:00:39"																			
19	17,"206.19","2.96","609.5","91.63","0.71","2025-06-29 15:15:04"																			
20	18,"214.08","3.1","664.47","91.77","0.66","2025-06-29 15:30:08"																			
21	19,"213.89","3.19","673.83","91.92","0.69","2025-06-29 15:45:51"																			
22	20,"216.78","3.43","744.33","91.86","0.68","2025-06-29 16:00:39"																			
23	21,"218.72","3.85","841.01","91.83","0.65","2025-06-29 16:15:46"																			
24	22,"219.89","3.24","712.79","91.85","0.56","2025-06-29 16:30:32"																			
25	23,"199.93","2.54","507.83","91.92","0.71","2025-06-29 16:45:39"																			
26	24,"208.44","3.61","753.22","91.94","0.67","2025-06-29 17:00:42"																			
27	25,"210.27","3.35","703.78","91.96","0.62","2025-06-30 11:00:50"																			
28	26,"200.78","3.74","751.78","91.97","0.71","2025-06-30 11:15:59"																			
29	27,"203.98","3.15","643.37","92.03","0.6","2025-06-30 11:30:05"																			
30	28,"217.19","2.31","501.7","92.05","0.74","2025-06-30 11:45:49"																			

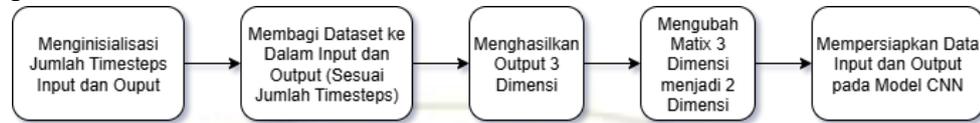
**Gambar 4. 10** Cuplikan *File CSV* setelah Diekspor dari MySQL

Proses peramalan menggunakan metode *Convolutional Neural Network* (CNN) dilakukan melalui platform Google Colaboratory. Kode program Python yang digunakan dapat dilihat pada Lampiran 2.



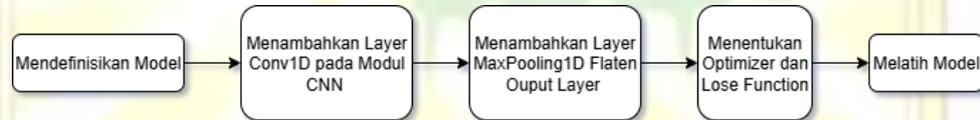
**Gambar 4. 11** Tahapan Proses Peramalan Menggunakan Metode CNN

Gambar 4.11 menampilkan keseluruhan tahapan proses peramalan dengan metode CNN, dimulai dari impor *library* dan pemuatan data *monitoring*. Proses persiapan data selanjutnya dijelaskan melalui alur kerja pada Gambar 4.12.



**Gambar 4. 12** Diagram Blok Proses Persiapan Data

Pada tahap ini, langkah pertama yang dilakukan adalah mengatur jumlah *time steps* yang akan digunakan untuk data *input* dan *output*, yaitu 25 untuk *input* dan 25 untuk *output*. Setelah itu, fungsi *split\_sequences* dipanggil untuk membagi *dataset* menjadi pasangan *input-output* berdasarkan jumlah *time steps* yang telah ditentukan. Hasil dari fungsi ini berupa *array* berdimensi tiga, di mana dimensi pertama menunjukkan jumlah sampel, dimensi kedua menunjukkan banyaknya *time steps*, dan dimensi ketiga menunjukkan jumlah fitur. Karena bentuk *output* masih dalam tiga dimensi, maka dilakukan proses *flattening* pada dimensi kedua dan ketiga untuk mengubahnya menjadi *array* dua dimensi. Selanjutnya, program mengambil nilai *n\_features* dari dimensi ketiga *array input* X. Sebagai langkah akhir, program mencetak nilai *n\_features*, *n\_output*, dan *y*.



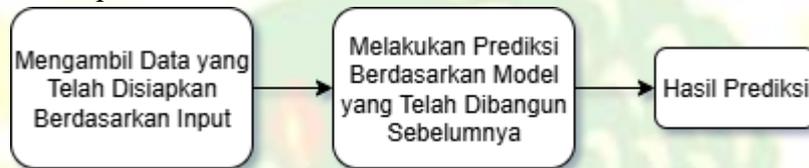
**Gambar 4. 13** Diagram proses kerja metode peramalan CNN

Pada tahap selanjutnya, proses peramalan dengan metode *Convolutional Neural Network* (CNN) dapat dilaksanakan setelah data *input* dan *output* selesai disiapkan. Parameter *n\_features* menunjukkan jumlah fitur dalam *dataset*, *n\_output* adalah jumlah keluaran setelah dilakukan proses *flattening*, dan *y* merupakan data *output* yang telah diubah menjadi *array* berdimensi dua. Alur lengkap tahapan ini dapat dilihat pada Gambar 4.13.

Langkah awal dalam membangun model dilakukan dengan mendefinisikan arsitekturnya menggunakan objek *Sequential* dari *library* Keras. Setelah itu, lapisan *Conv1D* ditambahkan, di mana parameter *filters* menunjukkan jumlah penyaring yang digunakan dan *kernel\_size* menetapkan ukuran dari penyaring tersebut. Fungsi pemacu yang diterapkan adalah *ReLU*. Sementara itu, *input\_shape* berfungsi untuk menetapkan dimensi masukan berdasarkan banyaknya langkah waktu (*time step*) dan jumlah fitur pada setiap langkah.

Selanjutnya, lapisan MaxPooling1D ditambahkan untuk mereduksi dimensi data dengan ukuran jendela yang ditentukan oleh parameter *pool\_size*. Setelah itu, lapisan *Flatten* digunakan untuk mengubah *output* dari bentuk *multi-dimensi* menjadi vektor satu dimensi. Model kemudian dilengkapi dengan lapisan *Dense*, dengan jumlah neuron sebanyak 50 dan fungsi pemicu ReLU. Terakhir, ditambahkan lapisan *output* dengan jumlah neuron sesuai dengan jumlah target keluaran yang diinginkan.

*Optimisasi* dilakukan dengan menggunakan algoritma Adam dan fungsi *loss* yang digunakan adalah *Mean Squared Error* (MSE). Proses pelatihan model dijalankan menggunakan data *input* X dan *output* y, dengan jumlah iterasi yang ditentukan oleh parameter *epochs*. Sementara itu, parameter *verbose* diatur ke nilai 0 untuk menonaktifkan tampilan proses pelatihan pada layar. Seluruh proses ini diimplementasikan melalui kode yang tercantum pada Lampiran 1.



**Gambar 4. 14** Diagram Alir Proses Pembuatan Prediksi

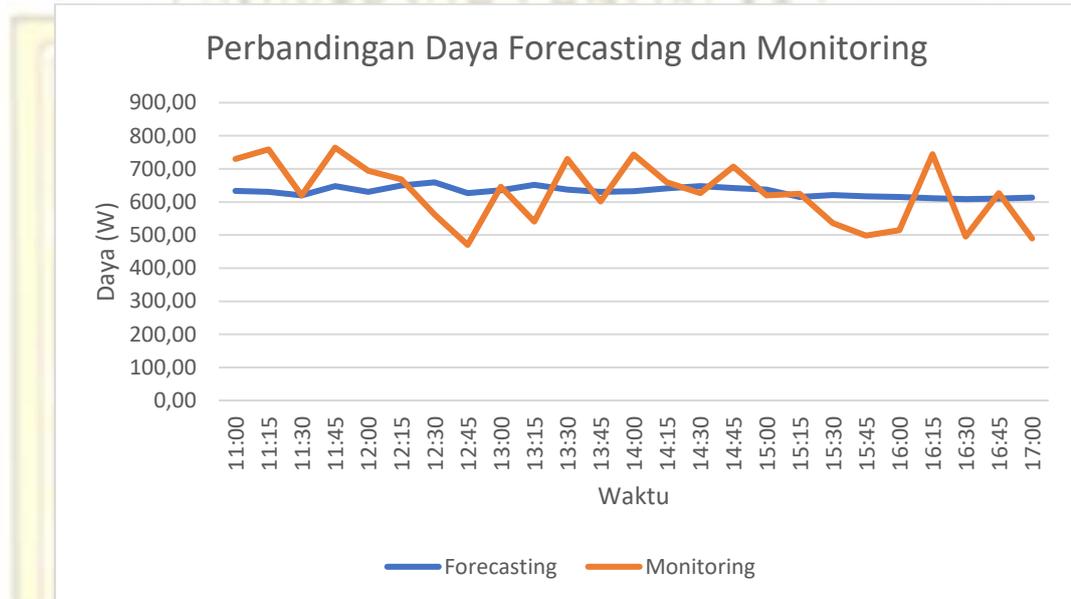
Langkah berikutnya dalam proses peramalan adalah melakukan prediksi, seperti yang ditampilkan pada Gambar 4.14. Program pada Lampiran A.2 akan terlebih dahulu mengambil 720 data terakhir dari *dataset* pelatihan menggunakan fungsi *tail()*, lalu menyimpannya dalam variabel *x\_input*. Data ini kemudian diubah bentuknya menggunakan fungsi *reshape()* agar sesuai dengan format *input* yang dibutuhkan oleh model CNN yang telah dibuat. Setelah itu, model yang sudah dilatih digunakan untuk memproses data tersebut dan menghasilkan prediksi, yang disimpan dalam variabel *yhat*. Terakhir, hasil dari prediksi akan ditampilkan melalui fungsi *print()*. Hasil akhir dari proses prediksi ini dapat dilihat pada Tabel 4.1.

**Tabel 4. 1** Hasil Peramalan *Monitoring* Genset

Daya	Getaran	Suhu
633,59	91,94	0,64
630,48	91,93	0,66
619,50	91,90	0,65
647,63	91,98	0,65
630,95	91,94	0,65

#### 4.5. Analisis Hasil Peramalan Jangka Pendek Daya *Monitoring* Genset Menggunakan Metode CNN

Pada Gambar 4.15 ditampilkan grafik perbandingan hasil peramalan daya jangka pendek menggunakan metode *Convolutional Neural Network* (CNN) dengan data daya aktual. Perbandingan ini menunjukkan sejauh mana akurasi model CNN dalam memprediksi *output* daya genset berdasarkan data historis. Adapun nilai-nilai detail hasil prediksi dan daya aktual dapat dilihat pada Tabel 4.2.



**Gambar 4. 15** Grafik Perbandingan Daya Aktual Genset dan Hasil Peramalan Menggunakan Metode CNN

**Tabel 4. 2** Tabel Komparasi Daya Prediksi dan Daya Aktual Genset

<b>Waktu</b>	<b>Daya Forecasting</b>	<b>Daya Monitoring</b>
11:00	633,59	730,32
11:15	630,48	758,81
11:30	619,50	620,06
11:45	647,63	764,20
12:00	630,95	694,65
12:15	650,16	668,30
12:30	659,35	562,36
12:45	626,67	470,16
13:00	635,25	645,88
13:15	651,50	540,31
13:30	637,21	729,77
13:45	630,32	601,30
14:00	632,69	743,78
14:15	641,43	659,06
14:30	647,58	626,65
14:45	641,71	706,46
15:00	637,15	620,16
15:15	614,99	624,40
15:30	620,89	536,01
15:45	617,27	498,20
16:00	615,12	514,59
16:15	610,81	744,80
16:30	608,54	495,15
16:45	610,42	626,55
17:00	613,62	489,86

Evaluasi terhadap hasil peramalan yang dilakukan menggunakan metode CNN dilakukan dengan menghitung nilai *Mean Squared Error* (MSE), *Mean Absolute Percentage Error* (MAPE), *Mean Percentage Error* (MPE). Evaluasi ini dilakukan dengan cara mengumpulkan serta membandingkan data daya hasil peramalan dengan data daya aktual yang telah diperoleh.

1. MSE

Untuk memperoleh nilai *Mean Squared Error* (MSE), digunakan rumus sebagai berikut dengan  $n = 25$ , di mana  $n$  merupakan jumlah data yang digunakan dalam perhitungan.

$$MSE = \frac{\sum_{t=1}^n (y_t - \tilde{y}_t)^2}{n}$$

$$MSE = \frac{1}{25} ((730,32 - 633,59)^2 + \dots + (489,86 - 613,62)^2)$$

$$MSE = \frac{1}{25} ((96,73)^2 + \dots + (-123,76)^2)$$

$$MSE = \frac{1}{25} (9.356,69 + \dots + 15.316,54)$$

$$MSE = \frac{195124,43}{25}$$

$$MSE = 7804,98$$

2. MAPE

Untuk memperoleh nilai *Mean Absolute Percentage Error* (MAPE), digunakan rumus sebagai berikut dengan  $n = 25$ , di mana  $n$  merupakan jumlah data yang digunakan dalam perhitungan.

$$MAPE = \frac{\sum_{t=1}^n \frac{|y_t - \tilde{y}_t|}{y_t}}{n} \times 100\%$$

$$MAPE = \frac{\frac{|730,32 - 633,59|}{730,32} + \dots + \frac{|489,86 - 613,62|}{489,86}}{25} \times 100\%$$

$$MAPE = \frac{\frac{96,73}{730,32} + \dots + \frac{123,76}{489,86}}{25} \times 100\%$$

$$MAPE = \frac{0,13 + \dots + 0,25}{25} \times 100\%$$

$$MAPE = \frac{3,10}{25} \times 100\%$$

$$MAPE = 0,124 \times 100\%$$

$$MAPE = 12,4\%$$

### 3. MPE

Untuk memperoleh nilai *Mean Percentage Error* (MPE), digunakan rumus sebagai berikut dengan  $n = 25$ , di mana  $n$  merupakan jumlah data yang digunakan dalam perhitungan.

$$MPE = \frac{\sum_{t=1}^n \frac{y_t - \tilde{y}_t}{y_t}}{n} \times 100\%$$

$$MPE = \frac{\frac{730,32 - 633,59}{730,32} + \dots + \frac{489,86 - 613,62}{489,86}}{25} \times 100\%$$

$$MPE = \frac{\frac{96,73}{730,32} + \dots + \frac{-123,76}{489,86}}{25} \times 100\%$$

$$MPE = \frac{0,13 + \dots + (-0,25)}{25} \times 100\%$$

$$MPE = \frac{-0,69}{25} \times 100\%$$

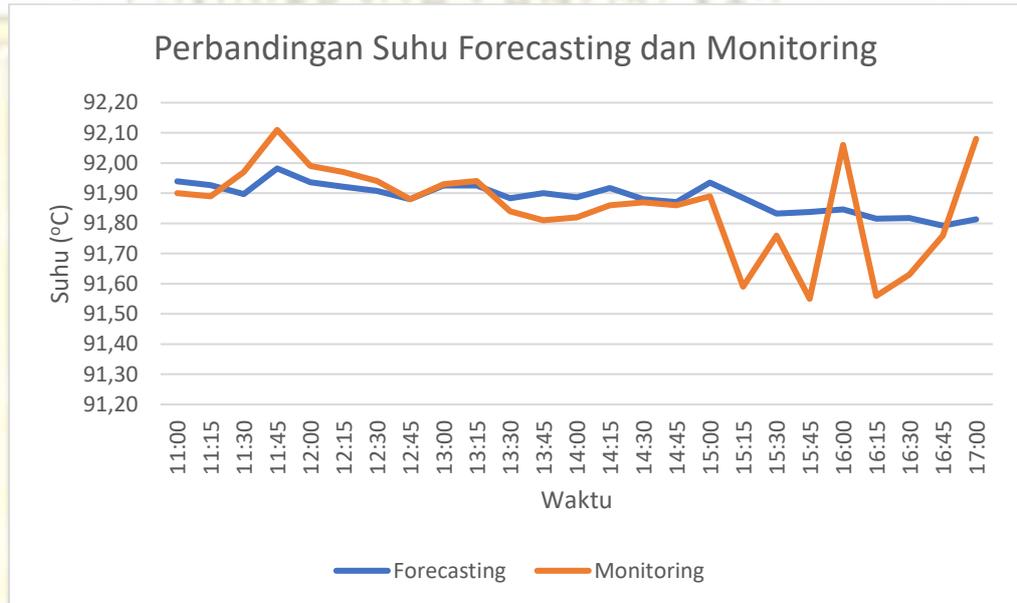
$$MPE = -0,276 \times 100\%$$

$$MPE = -2,76\%$$

Hasil peramalan daya menunjukkan nilai *Mean Squared Error* (MSE) sebesar 7.804,98, yang berarti masih ada perbedaan cukup besar antara data asli dan hasil prediksi. Nilai *Mean Absolute Percentage Error* (MAPE) sebesar 12,4% menunjukkan tingkat kesalahan yang masih bisa diterima. Sementara itu, *Mean Percentage Error* (MPE) sebesar -2,76% menunjukkan bahwa model sedikit melebihi hasil prediksi dibandingkan nilai sebenarnya. Secara keseluruhan, model sudah bekerja cukup baik, meskipun masih bisa ditingkatkan akurasi.

#### 4.6. Analisis Hasil Peramalan Jangka Pendek Suhu *Monitoring* Genset Menggunakan Metode CNN

Gambar 4.16 menyajikan grafik perbandingan antara hasil prediksi suhu jangka pendek menggunakan metode *Convolutional Neural Network* (CNN) dengan data suhu aktual. Grafik tersebut memberikan ilustrasi mengenai tingkat akurasi model CNN dalam memprediksi suhu genset berdasarkan data historis yang telah dikumpulkan. Informasi rinci mengenai nilai prediksi dan data aktual secara lengkap dapat dilihat pada Tabel 4.3.



**Gambar 4. 16** Grafik Perbandingan Suhu Genset Aktual dengan Hasil Prediksi Menggunakan Metode CNN

**Tabel 4. 3** Tabel Hasil Perbandingan Suhu Aktual dan Suhu Hasil Prediksi Genset

<b>Waktu</b>	<b>Suhu Forecasting</b>	<b>Suhu Monitoring</b>
11:00	91,94	91,90
11:15	91,93	91,89
11:30	91,90	91,97
11:45	91,98	92,11
12:00	91,94	91,99
12:15	91,92	91,97
12:30	91,91	91,94
12:45	91,88	91,88
13:00	91,93	91,93
13:15	91,93	91,94
13:30	91,88	91,84
13:45	91,90	91,81
14:00	91,89	91,82
14:15	91,92	91,86
14:30	91,88	91,87
14:45	91,87	91,86
15:00	91,93	91,89
15:15	91,88	91,59
15:30	91,83	91,76
15:45	91,84	91,55
16:00	91,85	92,06
16:15	91,82	91,56
16:30	91,82	91,63
16:45	91,79	91,76
17:00	91,81	92,08

Evaluasi hasil peramalan menggunakan metode *Convolutional Neural Network* (CNN) dilakukan dengan menghitung nilai *Mean Squared Error* (MSE), *Mean Absolute Percentage Error* (MAPE), dan *Mean Percentage Error* (MPE). Proses evaluasi ini dilakukan dengan membandingkan data suhu hasil prediksi dengan data suhu aktual yang telah dikumpulkan sebelumnya.

1. MSE

Untuk menghitung nilai *Mean Squared Error* (MSE), digunakan rumus berikut dengan  $n = 25$ , di mana  $n$  merupakan jumlah data yang digunakan dalam proses evaluasi.

$$MSE = \frac{\sum_{t=1}^n (y_t - \tilde{y}_t)^2}{n}$$

$$MSE = \frac{1}{25} ((91,90 - 91,94)^2 + \dots + (92,08 - 91,81)^2)$$

$$MSE = \frac{1}{25} ((-0,04)^2 + \dots + (0,27)^2)$$

$$MSE = \frac{1}{25} (0,0016 + \dots + 0,0729)$$

$$MSE = \frac{0,44}{25}$$

$$MSE = 0,02$$

2. MAPE

Untuk menghitung nilai *Mean Absolute Percentage Error* (MAPE), digunakan rumus berikut ini dengan  $n = 25$ , di mana  $n$  menunjukkan jumlah data yang digunakan dalam analisis.

$$MAPE = \frac{\sum_{t=1}^n \frac{|y_t - \tilde{y}_t|}{y_t}}{n} \times 100\%$$

$$MAPE = \frac{\frac{|91,90 - 91,94|}{91,90} + \dots + \frac{|92,08 - 91,81|}{92,08}}{25} \times 100\%$$

$$MAPE = \frac{\frac{0,04}{91,90} + \dots + \frac{0,27}{92,08}}{25} \times 100\%$$

$$MAPE = \frac{4,35 \times 10^{-4} + \dots + 29,32 \times 10^{-4}}{25} \times 100\%$$

$$MAPE = \frac{2,57 \times 10^{-2}}{25} \times 100\%$$

$$MAPE = 1,029 \times 10^{-3} \times 100\%$$

$$MAPE = 0,1029\%$$

### 3. MPE

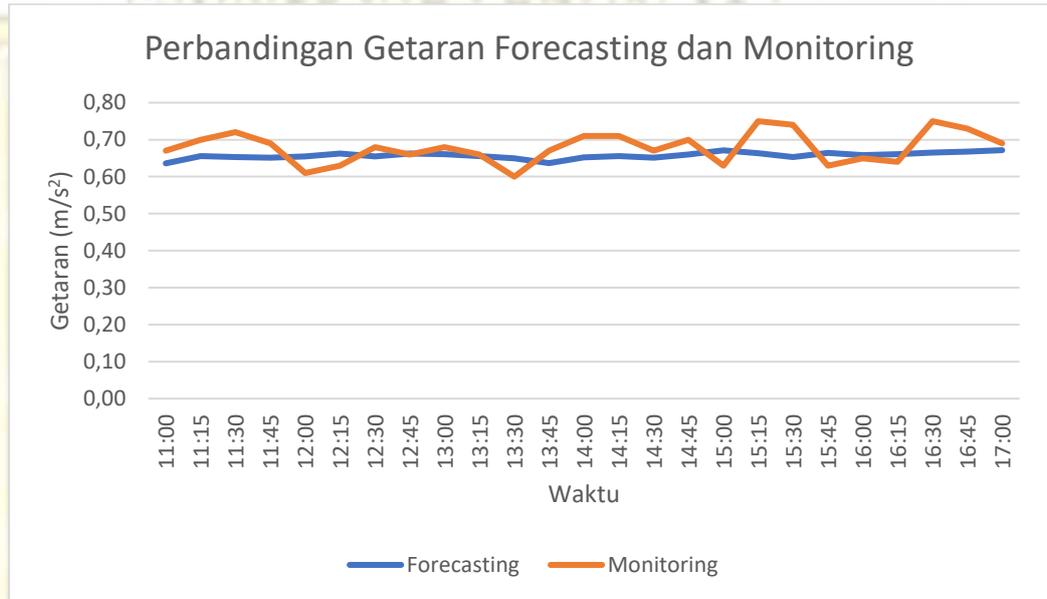
Untuk menghitung nilai *Mean Percentage Error* (MPE), digunakan rumus berikut ini dengan  $n = 25$ , yang merupakan jumlah total data yang digunakan dalam proses perhitungan.

$$\begin{aligned}MPE &= \frac{\sum_{t=1}^n \frac{y_t - \tilde{y}_t}{y_t}}{n} \times 100\% \\MPE &= \frac{\frac{91,90 - 91,94}{91,90} + \dots + \frac{92,08 - 91,81}{92,08}}{25} \times 100\% \\MPE &= \frac{\frac{-4 \times 10^{-2}}{9190 \times 10^{-2}} + \dots + \frac{27 \times 10^{-2}}{9208 \times 10^{-2}}}{25} \times 100\% \\MAPE &= \frac{\frac{-4}{9190} + \dots + \frac{27}{9208}}{25} \times 100\% \\MPE &= \frac{-4,35 \times 10^{-4} + \dots + 29,32 \times 10^{-4}}{25} \times 100\% \\MPE &= \frac{-7,62 \times 10^{-3}}{25} \times 100\% \\MPE &= -3,05 \times 10^{-4} \times 100\% \\MPE &= -0,000305\%\end{aligned}$$

Hasil peramalan suhu menghasilkan nilai MSE sebesar 0,02, yang menunjukkan bahwa perbedaan antara data aktual dan hasil prediksi sangat kecil. Nilai MAPE sebesar 0,1029% mencerminkan tingkat kesalahan yang sangat rendah, sedangkan nilai MPE sebesar  $-0,000305\%$  mengindikasikan bahwa prediksi model hampir sepenuhnya sesuai dengan data sebenarnya. Secara keseluruhan, model ini terbukti sangat akurat dan stabil dalam melakukan peramalan suhu.

#### 4.7. Analisis Hasil Peramalan Jangka Pendek Getaran *Monitoring* Genset Menggunakan Metode CNN

Gambar 4.17 menunjukkan grafik perbandingan antara hasil prediksi getaran jangka pendek menggunakan metode *Convolutional Neural Network* (CNN) dan data getaran aktual. Grafik ini digunakan untuk mengevaluasi seberapa akurat model CNN dalam memperkirakan nilai getaran genset berdasarkan data sebelumnya. Nilai-nilai lengkap dari hasil prediksi dan data aktual disajikan pada Tabel 4.4.



**Gambar 4. 17** Grafik Perbandingan Getaran Genset Aktual dan Prediksi Menggunakan Metode CNN

**Tabel 4. 4** Tabel Perbandingan Getaran Aktual dan Prediksi Genset

<b>Waktu</b>	<b>Suhu Forecasting</b>	<b>Suhu Monitoring</b>
11:00	0,64	0,67
11:15	0,66	0,70
11:30	0,65	0,72
11:45	0,65	0,69
12:00	0,65	0,61
12:15	0,66	0,63
12:30	0,65	0,68
12:45	0,66	0,66
13:00	0,66	0,68
13:15	0,66	0,66
13:30	0,65	0,60
13:45	0,64	0,67
14:00	0,65	0,71
14:15	0,66	0,71
14:30	0,65	0,67
14:45	0,66	0,70
15:00	0,67	0,63
15:15	0,66	0,75
15:30	0,65	0,74
15:45	0,66	0,63
16:00	0,66	0,65
16:15	0,66	0,64
16:30	0,66	0,75
16:45	0,67	0,73
17:00	0,67	0,69

Evaluasi hasil peramalan menggunakan metode *Convolutional Neural Network* (CNN) dilakukan dengan menghitung nilai *Mean Squared Error* (MSE), *Mean Absolute Percentage Error* (MAPE), dan *Mean Percentage Error* (MPE). Proses evaluasi ini dilakukan dengan membandingkan data suhu hasil prediksi dengan data suhu aktual yang telah dikumpulkan sebelumnya.

1. MSE

Untuk menentukan nilai *Mean Squared Error* (MSE), digunakan rumus berikut dengan  $n = 25$ , di mana  $n$  merepresentasikan jumlah data yang digunakan dalam proses evaluasi.

$$MSE = \frac{\sum_{t=1}^n (y_t - \tilde{y}_t)^2}{n}$$

$$MSE = \frac{1}{25} ((0,67 - 0,64)^2 + \dots + (0,69 - 0,67)^2)$$

$$MSE = \frac{1}{25} ((0,03)^2 + \dots + (0,02)^2)$$

$$MSE = \frac{1}{25} (0,0009 + \dots + 0,0004)$$

$$MSE = \frac{0,05}{25}$$

$$MSE = 0,002$$

2. MAPE

Nilai *Mean Absolute Percentage Error* (MAPE) dapat dihitung menggunakan rumus berikut, dengan  $n = 25$ , yang merupakan jumlah data yang dipakai dalam proses analisis.

$$MAPE = \frac{\sum_{t=1}^n \frac{|y_t - \tilde{y}_t|}{y_t}}{n} \times 100\%$$

$$MAPE = \frac{\frac{|0,67 - 0,64|}{0,67} + \dots + \frac{|0,61 - 0,65|}{0,61} + \dots + \frac{|0,69 - 0,67|}{0,69}}{25}$$

$$\times 100\%$$
$$MAPE = \frac{\frac{0,03}{0,67} + \dots + \frac{0,04}{0,61} + \dots + \frac{0,02}{0,69}}{25} \times 100\%$$

$$MAPE = \frac{0,04 + \dots + 0,07 + \dots + 0,03}{25} \times 100\%$$

$$MAPE = \frac{1,46}{25} \times 100\%$$

$$MAPE = 0,0586 \times 100\%$$

$$MAPE = 5,86\%$$

### 3. MPE

Rumus untuk menghitung nilai *Mean Percentage Error* (MPE) digunakan dengan  $n = 25$ , di mana  $n$  merepresentasikan jumlah keseluruhan data yang digunakan dalam perhitungan.

$$MPE = \frac{\sum_{t=1}^n \frac{y_t - \tilde{y}_t}{y_t}}{n} \times 100\%$$

$$MPE = \frac{\frac{0,67 - 0,64}{0,67} + \dots + \frac{0,61 - 0,65}{0,61} + \dots + \frac{0,69 - 0,67}{0,69}}{25} \times 100\%$$

$$MAPE = \frac{\frac{0,03}{0,67} + \dots + \frac{-0,04}{0,61} + \dots + \frac{0,02}{0,69}}{25} \times 100\%$$

$$MPE = \frac{0,04 + \dots + (-0,07) + \dots + 0,03}{25} \times 100\%$$

$$MPE = \frac{0,71}{25} \times 100\%$$

$$MPE = 0,0284 \times 100\%$$

$$MPE = 2,84\%$$

Hasil peramalan getaran menghasilkan nilai MSE sebesar 0,002, yang mengindikasikan bahwa perbedaan antara prediksi dan data aktual sangat kecil. Nilai MAPE sebesar 5,86% mencerminkan tingkat kesalahan yang relatif rendah, sementara nilai MPE sebesar 2,84% menunjukkan bahwa model cenderung sedikit mengestimasi di bawah nilai aktual. Secara keseluruhan, model CNN menunjukkan kinerja yang cukup akurat dalam memprediksi nilai getaran.

## BAB V SIMPULAN DAN SARAN

### 5.1. Simpulan

Berdasarkan seluruh hasil penelitian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

1. Peramalan Daya Genset Menggunakan CNN

Model *Convolutional Neural Network* memprediksi daya genset dengan MSE 7.804,98, MAPE 12,4%, dan MPE -2,76%, menunjukkan akurasi yang cukup namun masih bisa ditingkatkan.

2. Peramalan Suhu Genset Menggunakan CNN

Hasil peramalan suhu menunjukkan MSE 0,02, MAPE 0,1029%, dan MPE -0,000305%, menandakan model sangat akurat dan stabil.

3. Peramalan Getaran Genset Menggunakan CNN

Model CNN pada variabel getaran menghasilkan MSE 0,002, MAPE 5,86%, dan MPE 2,84%, yang menunjukkan prediksi cukup akurat dengan kecenderungan sedikit lebih rendah dibandingkan nilai sebenarnya.

### 5.2. Saran

Berdasarkan temuan dan keterbatasan dalam penelitian ini, sejumlah saran untuk pengembangan selanjutnya adalah sebagai berikut:

1. Peningkatan Model Peramalan

Eksperimen dengan arsitektur *deep learning* lain, seperti LSTM atau GRU, atau menggunakan kombinasi beberapa model (*ensemble*), dapat menjadi alternatif untuk menurunkan nilai MSE dan MAPE pada peramalan daya guna meningkatkan akurasi model.

2. Pelebaran Cakupan Data

Menambah durasi data histori (lebih dari 3 hari) serta meningkatkan frekuensi sampling dapat memperkaya variasi data latih dan membantu model dalam menghasilkan prediksi yang lebih akurat dan andal.

3. Integrasi *Real-Time Forecasting*

Mengimplementasikan *pipeline* pemrosesan otomatis memungkinkan hasil prediksi diperoleh secara *real-time* langsung dari *database* atau *stream* MQTT, sehingga sistem dapat memberikan rekomendasi pemeliharaan

## DAFTAR PUSTAKA

- [1] J. Brier dan lia dwi jayanti, "No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析Title," vol. 21, no. 1, hlm. 1–9, 2020.
- [2] J. Marqusee, C. Schultz, dan D. Robyn, "Power Begins at Home: Assured Energy for U.S. Military Bases," *Noblis*, 2017.
- [3] E. Haun, "Predictive maintenance," *Offshore Engineer*, vol. 44, no. 5, hlm. 725–729, 2019.
- [4] P. Kung, L. Wang, dan M. I. Comanici, "Stator end winding vibration and temperature rise monitoring," *2011 Electrical Insulation Conference, EIC 2011*, no. June, hlm. 10–14, 2011, doi: 10.1109/EIC.2011.5996105.
- [5] P. Studi, T. Listrik, J. Teknik, E. Fakultas, dan U. M. Makassar, "Rancang bangun instalasi genset pada fakultas teknik universitas muhammadiyah makassar," 2015.
- [6] W. Principle dan D. Generator, "Working Principle of Diesel Generator + Diagram linqip.com/blog/working-principle-of-diesel-generator," 2021.
- [7] M. Mobarra, M. Rezkallah, dan A. Ilinca, "Variable Speed Diesel Generators: Performance and Characteristic Comparison," *Energies (Basel)*, vol. 15, no. 2, 2022, doi: 10.3390/en15020592.
- [8] A. Sulthoni dan B. Suprianto, "Rancang Bangun Sistem Pendeteksi Vibrasi pada Motor Sebagai Indikator Pengaman Terhadap Perubahan Beban Menggunakan Sensor Accelerometer GY-521 MPU 6050 Berbasis Arduino Uno," *Jurnal Teknik Elektro*, vol. 7, no. 3, hlm. 147–155, 2018, [Daring]. Tersedia pada:  
<https://jurnalmahasiswa.unesa.ac.id/index.php/JTE/article/view/25095>
- [9] *Mechanical vibration : evaluation of machine vibration by measurements on non-rotating parts. Part 5, Machine sets in hydraulic power generating and pumping plants*. BSI, 2001.
- [10] M. T. Iwan Setiawan, S.T., "Buku Ajar Sensor dan Tranduser," *Semarang, Universitas Diponegoro*, hlm. 1–49, 2011.
- [11] S. Ginting, J. Welman, S. Electrical, E. Dept, I. Bukhori, dan E. R. Kaburuan, "Monitoring of Electrical Output Power-Based Internet of Things for Micro-Hydro Power Plant."

- [12] A. Ramelan, F. Adriyanto, B. A. C. Hermanu, M. H. Ibrahim, J. S. Saputro, dan O. Setiawan, "IoT Based Building Energy Monitoring and Controlling System Using LoRa Modulation and MQTT Protocol," *IOP Conf Ser Mater Sci Eng*, vol. 1096, no. 1, hlm. 012069, Mar 2021, doi: 10.1088/1757-899x/1096/1/012069.
- [13] "Monitoring Daya Listrik Berbasis IoT (Internet of Things)."
- [14] R. Khwanrit, S. Kittipiyakul, J. Kudtongngam, dan H. Fujita, "Accuracy Comparison of Present Low-cost Current Sensors for Building Energy Monitoring."
- [15] S. Tandel, P. Godbole, M. Malgaonkar, R. Gaikwad, dan R. Padaya, "An Improved Health Monitoring System using IOT," *SSRN Electronic Journal*, 2022, doi: 10.2139/ssrn.4109039.
- [16] M. I. Hafidhin, A. Saputra, Y. Rahmanto, dan S. Samsugi, "Alat Penjemuran Ikan Asin Berbasis Mikrokontroler Arduino UNO," *Jurnal Teknik dan Sistem Komputer*, vol. 1, no. 2, hlm. 59–66, 2020, doi: 10.33365/jtikom.v1i2.210.
- [17] V. Pravalika dan C. R. Prasad, "Internet of Things Based Home Monitoring and Device Control Using Esp32," no. July, 2019.
- [18] Y. Cheddadi, H. Cheddadi, F. Cheddadi, F. Errahimi, dan N. Es-sbai, "Design and implementation of an intelligent low-cost IoT solution for energy monitoring of photovoltaic stations," *SN Appl Sci*, vol. 2, no. 7, hlm. 1–11, 2020, doi: 10.1007/s42452-020-2997-4.
- [19] R. Y. Endra, A. Cucus, F. N. Afandi, dan M. B. Syahputra, "Model Smart Room Dengan Menggunakan Mikrokontroler Arduino Untuk Efisiensi Sumber Daya," *Explore: Jurnal Sistem informasi dan telematika*, vol. 10, no. 1, 2019, doi: 10.36448/jsit.v10i1.1212.
- [20] G. G. Maulana, R. Mada, dan R. R. Purba, "Automation Storage System Based On SCADA Using PLC CP1H and CP1L," *Jurnal Rekayasa Elektrika*, vol. 18, no. 3, hlm. 144–151, 2022, doi: 10.17529/jre.v18i3.26363.
- [21] H. M. Allam dan A. A. Chaudhri, "Internet of things: Extracting latest challenges and solutions," *Journal of Communications*, vol. 12, no. 9, hlm. 538–542, 2017, doi: 10.12720/jcm.12.9.538-542.
- [22] A. Gilchrist, *Industry 4.0*.
- [23] M. S. Abubakari dan Mashoedah, "The Internet of Things (IoT) as an Emerging Technological Solution for the Covid-19 Pandemic Mitigation: An Overview," *J Phys Conf Ser*, vol. 1737, no. 1, 2021, doi: 10.1088/1742-6596/1737/1/012003.

- [24] M. N. Al-Azam, D. Rizaludin, Y. S. Raharjo, dan A. Nugroho, "Message Queuing Telemetry Transport dalam Internet of Things menggunakan ESP-32," *Jurnal Media Informatika Budidarma*, vol. 3, no. 3, hlm. 159, 2019, doi: 10.30865/mib.v3i3.1160.
- [25] R. P. Pratama, "Sistem Monitoring dan Kendali AC melalui Aplikasi Node-RED," *Jurnal Fokus Elektroda*, vol. 7, no. 3, hlm. 162–168, 2022.
- [26] M. Fikry, "Buku Basis Data," *Angewandte Chemie International Edition*, 6(11), 951–952., hlm. 5–24, 1967.
- [27] A. Saputra, "Manajemen Basis Data Mysql Pada Situs FTP Lapan Bandung," *Jurnal Berita Dirgantara*, vol. 13, no. 4, hlm. 155–162, 2012.
- [28] D. D. Jantce TJ Sitinjak, . Maman, dan J. Suwita, "Analisa Dan Perancangan Sistem Informasi Administrasi Kursus Bahasa Inggris Pada Intensive English Course Di Ciledug Tangerang," *Insan Pembangunan Sistem Informasi dan Komputer (IPSIKOM)*, vol. 8, no. 1, 2020, doi: 10.58217/ipsikom.v8i1.164.
- [29] M. Ohyver, J. V. Moniaga, I. Sungkawa, B. E. Subagyo, dan I. A. Chandra, "The comparison firebase realtime database and MySQL database performance using wilcoxon signed-rank test," *Procedia Comput Sci*, vol. 157, hlm. 396–405, 2019, doi: 10.1016/j.procs.2019.08.231.
- [30] Muhammad Romzi dan B. Kurniawan, "Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma," *JTIM: Jurnal Teknik Informatika Mahakarya*, vol. 03, no. 2, hlm. 37–44, 2020.
- [31] A. Lusiana dan P. Yuliarty, "PENERAPAN METODE PERAMALAN (FORECASTING) PADA PERMINTAAN ATAP di PT X," *Industri Inovatif: Jurnal Teknik Industri*, vol. 10, no. 1, hlm. 11–20, 2020, doi: 10.36040/industri.v10i1.2530.
- [32] D. R. Indah dan E. Rahmadani, "Sistem Forecasting Perencanaan Produksi dengan Metode Single Eksponensial Smoothing pada Keripik Singkong Srikandi Di Kota Langsa," *Jurnal Penelitian Ekonomi Akuntansi (Jensi)*, vol. 2, no. 1, hlm. 10–18, 2019.
- [33] L. M. Rasdi Rere, M. I. Fanany, dan A. M. Arymurthy, "Metaheuristic Algorithms for Convolution Neural Network," *Comput Intell Neurosci*, vol. 2016, 2016, doi: 10.1155/2016/1537325.
- [34] R. H. Maryon dan C. C. Heasman, "THE ACCURACY OF PLUME TRAJECTORIES FORECAST USING THE U.K. METEOROLOGICAL OFFICE OPERATIONAL FORECASTING MODELS AND THEIR SENSITIVITY TO CALCULATION SCHEMES," 1988.

- [35] E. Sumando, "Pengembangan Metode Cash Forecasting Pemerintah: Studi Kasus Saldo Kas Pemerintah 2009 â€" 2011," *Kajian Ekonomi dan Keuangan*, vol. 2, no. 1, hlm. 70–93, 2018, doi: 10.31685/kek.v2i1.284.
- [36] M. Mulyadi, "Penelitian Kuantitatif Dan Kualitatif Serta Pemikiran Dasar Menggabungkannya," *Jurnal Studi Komunikasi dan Media*, vol. 15, no. 1, hlm. 128, 2013, doi: 10.31445/jskm.2011.150106.



## Lampiran 1 Program Monitoring (C++/Arduino)

```
#include <WiFi.h>

#include <PubSubClient.h>

#include <Wire.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>

// --- Konfigurasi WiFi dan MQTT ---
const char* ssid = "SSID WIFI";
const char* password = "Password WIFI";
const char* mqtt_server = "Server MQTT";
const int mqtt_port = Port MQTT;

WiFiClient espClient;
PubSubClient client(espClient);

// --- Sensor Tegangan (ZMPT101B) dan Arus (ACS712) ---
const int pinArus = 34;
const int pinTegangan = 35;

float kalibrasiTegangan = 0.1502732240437158;
float kalibrasiArus = 0.3;
float offsetADCArus = 2048.0; // Default, akan dikalibrasi saat setup()
```

```
// --- Sensor Getar dan Suhu (MPU6050) ---  
Adafruit_MPU6050 mpu;  
  
// --- Timer Publish ---  
unsigned long lastMsg = 0;  
const long interval = 5000;  
  
void setup_wifi() {  
  delay(100);  
  Serial.print("Menghubungkan ke WiFi: ");  
  Serial.println(ssid);  
  WiFi.begin(ssid, password);  
  
  int retry = 0;  
  while (WiFi.status() != WL_CONNECTED && retry < 20) {  
    delay(500);  
    Serial.print(".");  
    retry++;  
  }  
  
  if (WiFi.status() == WL_CONNECTED) {  
    Serial.println("\n✅ WiFi tersambung!");  
    Serial.print("IP ESP32: ");  
    Serial.println(WiFi.localIP());  
  
    IPAddress brokerIP;
```

```
if (WiFi.hostByName(mqtt_server, brokerIP)) {  
    Serial.print("✅ IP broker MQTT ditemukan: ");  
    Serial.println(brokerIP);  
} else {  
    Serial.println("❌ Gagal resolve IP broker!");  
}  
} else {  
    Serial.println("\n❌ Gagal terhubung ke WiFi.");  
}  
}  
  
void reconnect() {  
    while (!client.connected()) {  
        Serial.print("📶 Menghubungkan ke MQTT broker: ");  
        Serial.print(mqtt_server);  
        Serial.print(":");  
        Serial.println(mqtt_port);  
  
        if (client.connect("ESP32Client")) {  
            Serial.println("✅ Berhasil tersambung ke MQTT!");  
        } else {  
            Serial.print("❌ Gagal, rc=");  
  
            Serial.print(client.state());  
  
            Serial.println(" (rc -2 = broker tidak terjangkau)");  
  
            delay(5000);  
        }  
    }  
}
```

```
}  
}  
}
```

```
void kalibrasiOffset() {  
    long total = 0;  
    for (int i = 0; i < 500; i++) {  
        total += analogRead(pinArus);  
        delay(1);  
    }  
    offsetADCArus = total / 500.0;  
    Serial.print("Offset ADC Arus terkalibrasi: ");  
    Serial.println(offsetADCArus);  
}
```

```
float bacaTeganganRMS() {  
    long total = 0;  
    const int samples = 1000;  
    for (int i = 0; i < samples; i++) {  
        int offset = analogRead(pinTegangan) - 2048;  
        total += (long)offset * offset;  
        delayMicroseconds(100);  
    }
```

```
float vrms = sqrt((float)total / samples);  
Serial.print("ADC RMS: ");  
Serial.println(vrms);
```

```
float voltage = vrms * kalibrasiTegangan;

return voltage;
}

float bacaArus() {
    long total = 0;
    for (int i = 0; i < 100; i++) {
        total += analogRead(pinArus);
        delay(1);
    }
    float rata2 = total / 100.0;
    float tegangan = ((rata2 - offsetADCArus) / 4095.0) * 3.3;
    float arus = tegangan / kalibrasiArus;
    if (abs(arus) < 0.2) arus = 0;
    return abs(arus);
}

void setup() {
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, mqtt_port);

    delay(100);

    if (!mpu.begin()) {
        Serial.println("MPU6050 tidak terdeteksi. Periksa koneksi I2C.");
    }
}
```

```
while (1) delay(10);
}
Serial.println("MPU6050 terdeteksi!");

mpu.setAccelerometerRange(MPU6050_RANGE_2_G);
mpu.setGyroRange(MPU6050_RANGE_250_DEG);
mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);

kalibrasiOffset();
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  unsigned long now = millis();
  if (now - lastMsg > interval) {
    lastMsg = now;

    float voltage = bacaTeganganRMS();
    float current = bacaArus();

    float power = voltage * current;

    Serial.println("=== DATA SENSOR ===");
```

```
Serial.print("Tegangan (V): "); Serial.println(voltage);

Serial.print("Arus (A): "); Serial.println(current);

Serial.print("Daya (W): "); Serial.println(power);

sensors_event_t a, g, temp;
mpu.getEvent(&a, &g, &temp);

float suhu = temp.temperature;

float vibrasi = abs(sqrt(pow(a.acceleration.x, 2) + pow(a.acceleration.y, 2) +
pow(a.acceleration.z, 2)) - 9.80665);

Serial.print("Suhu (°C): "); Serial.println(suhu);
Serial.print("Getaran (m/s²): "); Serial.println(vibrasi);

client.publish("genset/tegangan", String(voltage).c_str());
client.publish("genset/arus", String(current).c_str());
client.publish("genset/daya", String(power).c_str());
client.publish("genset/suhu", String(suhu).c_str());
client.publish("genset/getaran", String(vibrasi).c_str());
Serial.println("Data dikirim ke MQTT.\n");
}
}
```

## Lampiran 2 Program Forecasting (Python/Google Colab)

```
# Import necessary libraries

import pandas as pd

from numpy import array
from sklearn.preprocessing import MinMaxScaler

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import Flatten

from keras.layers import Conv1D

from keras.layers import MaxPooling1D

from keras.callbacks import EarlyStopping, ReduceLROnPlateau

# ===== 1. BACA DATA CSV =====

# Load data

df = pd.read_csv("data_monitoring.csv")

df.set_index('id', inplace=True)

datatraining = df.copy() # pastikan df adalah DataFrame kamu

datatraining.drop(columns=['waktu', 'tegangan', 'arus'], inplace=True)
```

```

# ===== 2. SCALING DATA =====

scaler = MinMaxScaler()

datatraining_scaled = pd.DataFrame(
    scaler.fit_transform(datatraining),
    columns=datatraining.columns
)

# ===== 3. SPLIT SEQUENCES =====

def split_sequences(sequences, n_steps_in, n_steps_out):
    X, y = list(), list()
    for i in range(len(sequences)):
        end_ix = i + n_steps_in
        out_end_ix = end_ix + n_steps_out
        if out_end_ix > len(sequences):
            break
        seq_x = sequences.iloc[i:end_ix].values
        seq_y = sequences.iloc[end_ix:out_end_ix].values
        X.append(seq_x)
        y.append(seq_y)
    return array(X), array(y)

n_steps_in, n_steps_out = 25, 25

X, y = split_sequences(datatraining_scaled, n_steps_in, n_steps_out)

```

```
n_output = y.shape[1] * y.shape[2]
y = y.reshape((y.shape[0], n_output))
n_features = X.shape[2]
```

```
# ===== 4. MODEL CNN =====
```

```
model = Sequential()
model.add(Conv1D(filters=128, kernel_size=5, activation='relu',
input_shape=(n_steps_in, n_features)))
model.add(MaxPooling1D(pool_size=3))
model.add(Flatten())
model.add(Dense(32, activation='relu'))
model.add(Dense(n_output))
model.compile(optimizer='adam', loss='mse')
```

```
# ===== 5. CALLBACKS =====
```

```
callbacks = [
    EarlyStopping(monitor='loss', patience=20, restore_best_weights=True),
    ReduceLROnPlateau(monitor='loss', factor=0.5, patience=3)
]
```

```
# ===== 6. TRAINING =====
```

```
model.fit(X, y, epochs=1000, verbose=0, callbacks=callbacks)
```

```
# ===== 7. PREDIKSI =====  
  
x_input = datatraining_scaled.iloc[-n_steps_in:].values  
x_input = x_input.reshape((1, n_steps_in, n_features))  
yhat = model.predict(x_input, verbose=0)  
yhat = yhat.reshape((n_steps_out, n_features))  
  
# ===== 8. INVERSE SCALER KE SKALA ASLI =====  
  
yhat_df = pd.DataFrame(yhat, columns=datatraining.columns)  
yhat_inverse = scaler.inverse_transform(yhat_df)  
  
hasil_prediksi = pd.DataFrame(yhat_inverse, columns=["daya", "suhu",  
"getaran"])  
  
# ===== 9. SIMPAN KE Excel =====  
  
hasil_prediksi.to_excel("nama_file.xlsx", index=False)
```

# Cek Similarity

## TA 2 - Fadly Ihsan Andanny

 1810953001 Fadly Ihsan Andanny "Peramalan Kinerja Genset di Pondok Pesantren Harakatul Qur'an menggunakan Metode Convolut..."

---

### Document Details

Submission ID

trn:oid::3618:107176685

Submission Date

Aug 5, 2025, 2:46 PM GMT+7

Download Date

Aug 5, 2025, 2:52 PM GMT+7

File Name

TA 2 - Fadly Ihsan Andanny.docx

File Size

8.4 MB

99 Pages

15,388 Words

98,460 Characters

# 6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

## Filtered from the Report

- ▶ Bibliography
- ▶ Quoted Text
- ▶ Small Matches (less than 20 words)

## Exclusions

- ▶ 20 Excluded Sources

---

## Top Sources

- 5%  Internet sources
- 1%  Publications
- 4%  Submitted works (Student Papers)

---

## Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

## Top Sources

- 5% Internet sources
- 1% Publications
- 4% Submitted works (Student Papers)

## Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	scholar.unand.ac.id	2%
2	Student papers	andalas on 2025-04-17	<1%
3	Student papers	Fakultas Teknik on 2025-07-23	<1%
4	Internet	repository.its.ac.id	<1%
5	Internet	www.stmik-budidarma.ac.id	<1%
6	Internet	eprints.umpo.ac.id	<1%
7	Internet	eprints.uny.ac.id	<1%
8	Internet	dspace.uii.ac.id	<1%
9	Internet	openlibrarypublications.telkomuniversity.ac.id	<1%
10	Student papers	Institut Agama Islam Al-Zaytun Indonesia on 2024-08-21	<1%
11	Publication	Joko Susetyo, Endang Widuri Asih, Heharuman Raharjo. "Optimalisasi Jumlah Pro..."	<1%

12 Student papers

itera on 2025-06-30 <1%

---

13 Internet

repository.poltekbangplg.ac.id <1%

---

14 Student papers

Universitas Brawijaya on 2019-07-10 <1%

---

15 Internet

repository.upbatam.ac.id <1%