

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Kajian terhadap penelitian terdahulu perlu dilakukan untuk mengetahui implementasi teknologi IoT dan algoritma *Machine Learning Deep Learning* yang telah diterapkan pada sistem monitoring kualitas air, khususnya yang berbasis *TinyML* dan komputasi *edge*. Selain itu, kajian ini bertujuan untuk mengidentifikasi teknik optimisasi model seperti kuantisasi, pruning, dan *clustering* melalui *TensorFlow Lite* beserta kelebihan, keterbatasan memori perangkat, serta celah penelitian yang ada. Hasil kajian ini akan digunakan sebagai dasar dalam menentukan pendekatan kompresi model yang efisien agar dapat diimplementasikan secara optimal pada mikrokontroler dengan sumber daya terbatas. Jurnal terdahulu bisa dilihat pada Table 2.1

Tabel 2. 1 Teori Terkait

No	Judul	Metode	Hasil Evaluasi	Keterbatasan
1.	Advances and Challenges in TinyML-Based Water Trace Element Monitoring[26]	Tinjauan sistematis (PRISMA 2015-2025) terhadap 46 studi TinyML pada mikrokontroler (ESP32, Arduino) untuk pemantauan elemen air.	Akurasi klasifikasi laboratorium tinggi (75–99,8%). XGBoost menjadi algoritma yang paling banyak digunakan (33,33%).	Hanya 31% studi yang melakukan validasi lapangan; kendala RAM (<100 KB) pada 51,96% kasus; masalah sensor drift (68% studi).
2.	Penerapan Optimisasi Pruning dan	Ekstraksi ciri suara dengan MFCC dan	Mengurangi waktu inference	Optimasi model secara umum sangat

	Clustering TensorFlow Lite pada Embedded System untuk Sistem Speech to Text[27]	klasifikasi CNN. Model dioptimasi dengan teknik Pruning dan Clustering TFLite pada Raspberry Pi 4.	sebesar 93,3%, hemat RAM 17,1%. Evaluasi: Akurasi 0,97, Presisi 0,994, Recall 0,997, dan F1-score 0,997.	meningkatkan kecepatan dan memori, namun pada beberapa kasus ekstrem berisiko memicu sedikit penurunan akurasi.
3.	Water Quality Identification: Integrating IoT Sensors and Deep Learning for Near-Real-Time Water Quality Assessment[28]	IoT "Water-QI" (sensor suhu, TDS, pH, turbiditas, konduktivitas) dikombinasikan dengan arsitektur Gated Recurrent Unit (GRU).	Skenario per jam: GRU single-layer (64 unit) paling seimbang (RMSE 0,0281, 0,9820). Skenario per menit: versi 256 unit terbaik untuk komputasi edge.	Data berasal dari interpolasi (bukan data lapangan kontinu asli) sehingga kurang mencerminkan stochastic noise atau signal drift real-time.
4.	Analisis Performa TensorFlow Lite untuk IoT dengan ESP32 DEVKIT-C (Studi Kasus: Gambar Sampah Sungai)[29]	Pengujian performa TensorFlow Lite Micro pada ESP32 DevKit-C menggunakan variasi ukuran input gambar dan arsitektur model.	Metode Full Integer Quantization (Integer Only) terbukti menjadi metode kuantisasi terbaik untuk implementasi AI pada ESP32.	Ukuran input model dan total parameter berpengaruh signifikan terhadap performa komputasi dan batas kelayakan pasang (deployability) alat.
5.	Deep Learning for Water Quality[30]	Kajian pustaka (Review Paper) mengenai peluang, kekuatan, dan tantangan	DL terbukti efektif mengatasi kelangkaan data (data scarcity)	Model DL murni rentan menghasilkan prediksi yang tidak realistis secara fisik di

		metode Deep Learning (DL) konvensional vs model baru dalam memprediksi kualitas air.	melalui pengisian celah spasial-temporal dan penemuan pengetahuan baru.	luar kondisi pelatihan jika data latih terbatas atau kotor.
--	--	--	---	---

2.2 Landasan Teori

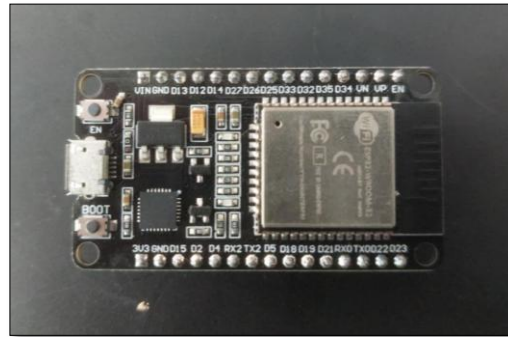
2.2.1 Internet Of Things

IoT yaitu kepanjangan dari *Internet of Things* merupakan sebuah media konektivitas pengiriman data yang memungkinkan sebuah perangkat terhubung ke internet dan bertukar data dengan perangkat lainnya. IoT bergantung pada teknologi komunikasi nirkabel untuk meyambungkan perangkat ke jaringan internet serta keterhubungan antar perangkat IoT dengan *platform* yang berbeda beda memerlukan standar dan protokol yang seragam untuk memastikan interoperabilitas yang baik. Konsep ini dikembangkan untuk mengintegrasikan berbagai sistem dan perangkat yang digunakan dalam berbagai bidang seperti pertanian, industri, transportasi, kesehatan, dan lain lainnya[31].

2.2.2 NodeMCU ESP32

NodeMCU ESP32 adalah platform mikrokontroler sumber terbuka yang telah mengintegrasikan modul *Wi-Fi* di dalamnya untuk mendukung konektivitas *Internet of Things* (IoT)[32]. Perangkat ini berfungsi sebagai unit pemrosesan pusat sekaligus media transmisi data hasil pembacaan sensor ke server atau platform digital secara

nirkabel[33][34]. Berikut adalah NodeMCU ESP32 dapat dilihat pada Gambar 2.1



Gambar 2. 1 *NodeMCU* ESP32

2.2.3 Sensor INA219

Sensor INA219 merupakan modul elektronik digital yang berfungsi untuk mengukur arus (*current*), tegangan (*voltage*), dan daya (*power*) secara simultan pada suatu rangkaian listrik. Modul ini mengintegrasikan sirkuit penguat analog dan pengubah analog-ke-digital (ADC) 12-bit internal untuk menghasilkan pembacaan data performa daya kelistrikan dengan tingkat akurasi yang tinggi. Sensor INA219 berperan sebagai unit pemantau kesehatan daya listrik (*power telemetry unit*) lokal yang mengirimkan parameter tegangan dan konsumsi arus riil secara berkala kepada ESP32, mendukung evaluasi efisiensi hemat daya pada perangkat keras yang diuji[35]. Berikut adalah Sensor INA219 dapat dilihat pada Gambar 2.2



Gambar 2. 2 Sensor INA219

2.2.4 Machine Learning

Machine Learning merupakan metode analisis data yang mengotomatisasi pembangunan model analitik, dimana komputer mampu belajar dan meningkatkan kemampuannya berdasarkan pengalaman tanpa perlu diprogram secara eksplisit. Teknologi ini bekerja dengan menggunakan algoritma statistik untuk mempelajari pola dalam data dengan tujuan meningkatkan akurasi serta efisiensi dalam setiap keputusan prediksi yang dihasilkan[36].

2.2.5 Tiny Machine Learning (TinyML)

Tiny Machine Learning merupakan sebuah kecerdasan buatan yang mengintegrasikan algoritma *Machine Learning* (ML) dan *Deep Learning* (DL) langsung ke dalam perangkat keras tertanam (*embedded system*) berdaya sangat rendah, seperti mikrokontroler (ESP32 atau Arduino). Perangkat *TinyML* umumnya beroperasi dengan batasan memori yang sangat ketat, sering kali memiliki RAM kurang dari 100 KB dan konsumsi daya di tingkat *miliwatt* (mW). Untuk menjalankan model pada arsitektur sekecil ini, *framework* khusus seperti *TensorFlow Lite for Microcontrollers* (TFLite Micro) digunakan untuk

mengeksekusi model yang telah dioptimasi tanpa memerlukan sistem operasi yang kompleks[37].

2.2.7 TensorFlow Lite

TensorFlow Lite (TFLite) adalah *framework* inferensi kecerdasan buatan berbasis sumber terbuka (*open-source*) yang dirancang untuk menjalankan model *Machine Learning* dan *Deep Learning* pada perangkat dengan sumber daya terbatas (*resource-constrained devices*), seperti mikrokontroler (misalnya ESP32) yang beroperasi tanpa sistem operasi. ini mampu mengeksekusi inferensi model jaringan saraf tiruan secara mandiri dengan kebutuhan memori statis yang sangat kecil, bahkan hanya dalam hitungan puluhan hingga ratusan *Kilobyte* (KB)[38].

2.2.7 Pruning (Pemangkasan)

Pruning merupakan teknik kompresi model yang dilakukan dengan mengeliminasi parameter atau bobot jaringan saraf yang tidak signifikan (bernilai mendekati nol). Teknik ini bertujuan untuk mengurangi redundansi parameter dan memperkecil ukuran file model[39].

2.2.8 Kuantisasi (*Quantization*)

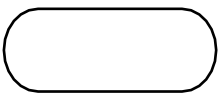
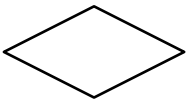
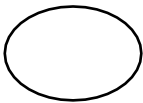
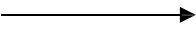
Kuantisasi merupakan proses mengkonversi representasi nilai *numeric* bobot (*weight*) dan aktivasi model dari format presisi tinggi menjadi format presisi yang lebih rendah. Pada umumnya, model *deep*

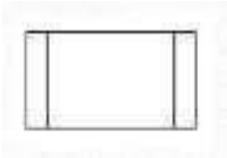
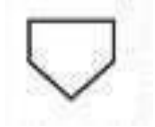
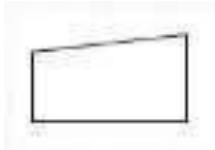
learning dilatih menggunakan format data *Floating Point* 32-bit (FP32). Melalui kuantisasi, nilai FP32 ini dipetakan ke dalam format bilangan bulat yang lebih kecil, seperti *integer* 8-bit (INT8)[40].

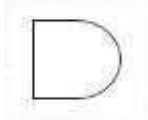
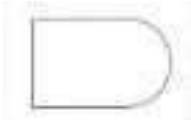
2.2.8 Flowchart

Flowchart merupakan gambaran berbentuk suatu grafik yang disertai langkah langkah dan urutan suatu prosedur dari suatu program. *Flowchart* dapat membantu proses analisis, perancangan dan pengkodean untuk memecahkan masalah kedalam bagian-bagian yang lebih kecil untuk pengoperasiannya. Tabel flowchart bias dilihat pada Tabel 2.2

Tabel 2. 2 Flowchart

No	Simbol	Nama	Fungsi
1.		Terminator	Simbol untuk permulaan (<i>start</i>) atau akhir (<i>stop</i>) dari suatu kegiatan.
2.		<i>Process</i>	Simbol menunjukkan proses atau langkah kerja.
3.		<i>Decision</i>	Simbol pemilihan proses berdasarkan kondisi yang ada.
4.		<i>Connector</i>	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar atau halaman yang sama.
5.		<i>Flow Direction</i>	Simbol yang digunakan untuk menghubungkan

			antara ssmbol yang satu dengan symbol yang lain. Simbol ini disebut juga <i>connection line</i> .
6.		<i>Input - Output</i>	Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
7.		<i>Predefined Process</i>	Merupakan simbol yang digunakan untuk menunjukkan sub-proses yang belum dijelaskan secara rinci dan akan diuraikan pada bagian lain
8.		<i>Connector (offpage)</i>	Merupakan simbol yang digunakan untuk menghubungkan simbol pada halaman yang berbeda.
9.		<i>Preparation Symbol</i>	Merupakan simbol untuk menunjukkan proses penyimpanan data ke dalam media penyimpanan.
10.		<i>Manual Input Symbol</i>	Merupakan simbol yang digunakan untuk menunjukkan proses <i>input</i> data secara manual menggunakan <i>keyboard</i> atau perangkat <i>input</i> .

11.		<i>Manual Operation Symbol</i>	Merupakan simbol yang digunakan untuk menunjukkan kegiatan atau proses yang dilakukan komputer atau sistem. Simbol ini biasanya menggambarkan aktivitas.
12.		<i>Display Symbol</i>	Merupakan simbol yang digunakan untuk menunjukkan penggunaan perangkat <i>output</i> , seperti monitor, printer, atau perangkat lainnya.
13.		<i>Delay Symbol</i>	Merupakan simbol yang digunakan untuk menunjukkan proses penundaan (<i>delay</i>) atau menunggu dalam suatu alur sistem.