BAB I PENDAHULUAN

1.1 Latar Belakang

Peningkatan polusi udara menjadi salah satu permasalahan lingkungan yang berdampak pada kesehatan manusia dan ekosistem. Sampai saat ini, belum ada solusi untuk masalah polusi udara. Salah satu cara untuk mengurangi pencemaran udara adalah dengan mengkategorikan kualitas udara. Melakukan pengukuran kualitas udara dalam ruangan secara manual sangat tidak efektif, jadi dalam penelitian ini dirancang sebuah sistem yang dapat mengumpulkan data tentang kualitas udara dalam ruangan secara *real time*[1].

Oleh karena itu, diperlukan sistem monitoring kualitas udara yang efektif dan efisien untuk mendeteksi perubahan kondisi udara secara real-time. *(Internet of Things) IoT* telah memungkinkan pengembangan sistem monitoring yang dapat mengumpulkan, mengirim, dan menganalisis data kualitas udara dengan lebih mudah. Dalam situasi di mana aplikasi IoT hanya perlu mengirimkan jumlah data yang kecil dan berdaya rendah secara jarak jauh, teknologi LoRa akan sesuai.[2] ,[3]

Sistem ini dirancang untuk mengakuisisi data dari berbagai sensor seperti suhu, kelembapan, dan gas menggunakan unit mikrokontroler, data di kirim ke platform Antares secara nirkabel melalui protokol LoRa, dan di implementasi kan pada platform *website* berbasis server local, agar dapat diakses dengan mudah oleh publik, *website* ini telah di-*hosting* secara daring. Platform web ini akan menyajikan data secara *real-time* melalui visualisasi yang intuitif seperti *gauge* dan grafik historis. Pengembangan sistem ini diharapkan dapat menjadi solusi monitoring kualitas udara yang efisien, terjangkau, dan dapat dikustomisasi sesuai kebutuhan lokal, sekaligus mendorong kesadaran masyarakat terhadap pentingnya menjaga kualitas udara.

Table 1 Jurnal Referensi

NO	Author	Year	Methods	Perfomance
1.	Sagala A, Hutapea S ...[4]	2024	LoRaWAN LGT92 : gateway *(RAK7243 & LG308)*	membuktikan bahwa implementasi multi-gateway LoRaWAN dapat mengoptimalkan sistem pelacakan maritim.
2.	Bogdan Filip, Attila Simo...[5]	2020	LoRaWAN : Gateway LoRa (Cisco IXM, Kerlink)	Sistem ini terbukti dapat memberikan data real-time tentang kualitas udara, mendukung pengambilan keputusan terkait lingkungan dan kesehatan masyarakat.

3.	Rabeya Anzum..[6]	2019	LoRaWAN : Gateway (Cisco IXM, Kerlink) Sensor udara (NO2, SO2, O3, CO, PM1, PM2.5, PM10)	LoRaWAN adalah solusi efektif dan hemat biaya untuk pemantauan kualitas udara, meskipun masih memerlukan peningkatan dalam jangkauan dan penetrasi sinyal.
4.	Hendri Hananta Wahyu Wibowo, Zulkhairi and Lilis Kurniasari.[7]	2024	LoRaWAN: Mikrokontroler: ESP32 Devkit, Sensor: MQ-135 (CO2, NOx, NH3, benzena), MQ-2 (asap, LPG, metana, hidrogen) , Platform IoT: Blynk (untuk monitoring real-time melalui web dan aplikasi mobile)	Sistem monitoring berbasis IoT dan LoRaWAN menunjukkan keandalan tinggi dalam mendeteksi polusi udara secara real-time, dengan sensor yang akurat dan sistem notifikasi yang responsif
5.	St.Nurhayati Djabir,Muhammad Fadli Azis, Amin Kurnia[8]	2022	Mikrokontroler: Wemos D1,R1 Sensor: Limit switch Aktuator:Buzzer PlatformIoT:Antares AplikasiMobile:MIT Inventor	Sistem mampu bekerja dengan respon waktu rata-rata 1.6 detik untuk monitoring kondisi pintu dan 2.3 detik untuk sistem kontrol dan alarm
6.	Geubrina Rizka Utami Sinaga, Samsudin[9]	2021	Waterfall, UML, Framework Laravel, PHP, MySQL, Blackbox Testing	Sistem reservasi online untuk pemesanan tempat dan menu berhasil dibangun dan diterapkan, dengan pengujian *blackbox* menunjukkan semua fungsi berjalan valid.
7.	Bragiwibisana, Fikra Titan Syifa, Indah Permatasari[10]	2024	SistemMinimum:ESP8266 PlatformIoT:Antares InstrumenUkur: Osiloskop Pengujian: Berdasarkan variasi jarak objek (1 cm - 5 cm)	Sistem ini efektif dalam mendeteksi logam melalui platform Antares, di mana performanya membuktikan bahwa jarak dan kandungan logam objek sangat mempengaruhi nilai tegangan dan ADC yang dihasilkan.
8.	Fried Sinlae, Eko Irwanda, Zaky Maulana, Vicky Eka Syahputra[11]	2024	Arsitektur: MVC (Model-View-Controller) •Fitur Utama yang Digunakan:Eloquent ORM (Manajemen Database) Blade Templating Engine (Antarmuka Pengguna) Sistem Migrasi Database (Pengelolaan Skema) Integrasi Layanan Pihak Ketiga (API, Sistem Pembayaran, Email)	Laravel adalah framework yang andal dan efisien untuk pengembangan aplikasi web berbasis PHP, meskipun pemilihan antara Laravel dan framework lain tetap bergantung pada kebutuhan spesifik proyek.

9.	Agis Fauzy Alfy Hasani, Abdul Aziz Apriyanda, Nurul Falah, Dhea Aulia Agustin Sugiarto, Rahmi Maryam Nur Riswandi[12]	-	Database:MySQL Bahasa:SQL Tools: cPanel, Command Prompt (WAMP/XAMPP), phpMyAdmin	MySQL adalah solusi DBMS yang efektif dan *open source* untuk aplikasi web, meskipun kurang cocok untuk menangani data dalam skala yang sangat besar dan dukungan pemrogramannya lebih terfokus pada sisi *backend*.
10.	Rizky Fajar Ramadhan, Riki Mukhaiyar[13]	2020	Server:RaspberryPi (sebagai serverpublik Database: MySQL Interface: phpMyAdmin Aplikasi Klien: Android Studio (Java)	Penggunaan Raspberry Pi sebagai server publik untuk database MySQL merupakan solusi efektif dan hemat biaya untuk kontrol *smarthome* , meskipun performa sistem sangat bergantung pada kualitas dan kecepatan koneksi internet
11.	M.Mujiono, Adimas KetutNalendra, DimasHafidhFauzi, Niswatul Karromah[14]	2023	Menggunakan IoT dengan Arduino, sensor suhu (DHT11), dan sensor gas amonia (MQ-135). Data dikirim melalui modul WiFi ESP8266 dengan protokol HTTP ke *website* monitoring yang dibangun menggunakan *framework* Laravel dan database MySQL. Pengembangan sistem memakai metode *waterfall*	Alat berhasil terhubung dengan *access point* dan *website*. Sensor terbukti dapat mendeteksi suhu dan kadar gas amonia. *Website* yang dibuat dapat menyimpan dan menampilkan data hasil pembacaan sensor sesuai dengan perencanaan
12.	Rendra Dwi Bima Sakti, Sri Lestanti, Saiful Nur Budiman[15]	2024	Menggunakan metode pengembangan *prototyping*. Pembangunan *dashboard* menggunakan *framework* Laravel dan VueJS dengan mengadopsi metrik KPI *dashboard sales*. Pengujian sistem dilakukan menggunakan *black box testing* dan evaluasi kualitas informasi menggunakan model DeLone and McLean.	*Dashboard* berhasil diimplementasikan. Pengujian *black box testing* menghasilkan tingkat keberhasilan 91%. Evaluasi kualitas informasi menggunakan model DeLone and McLean memperoleh hasil rata-rata 89.28%, yang menandakan *dashboard* layak digunakan

13.	Delly Yusar Akbar, Yoga Alif Kurnia[16]	2024	Sistem ini menggunakan sensor MQ-7 untuk mendeteksi kadar karbon monoksida. Data diolah oleh mikrokontroler Wemos D1 R2 dan dikirim ke database MySQL. Sebuah website yang dibangun dengan *framework* Laravel digunakan untuk menampilkan data monitoring	Hasil pengujian sensor MQ-7 menunjukkan persentase kesalahan pengukuran antara 2,5% hingga 47,5%, dengan rata-rata kesalahan 10,4%.*Website* monitoring dapat berfungsi dengan baik dan menampilkan data selama terhubung dengan koneksi internet.
14.	Fried Sinlae, Eko Irwanda, Zaky Maulana, Vicky Eka Syahputra[17]	2024	Efektivitas *framework* Laravel dalam pengembangan website.Analisis berfokus pada kecepatan pengembangan,kemudahan integrasi dengan layanan pihak ketiga (sistem pembayaran, email, API sosial), serta performa dan skalabilitas aplikasi	Studi ini menyimpulkan bahwa Laravel dapat mempercepat waktu pengembangan hingga 30%. Integrasi dengan layanan pihak ketiga dinilai sangat mudah, dengan skor kepuasan pengembang 9/10 untuk sistem pembayaran. Dari segi performa, aplikasi Laravel menunjukkan waktu respon lebih cepat (150 ms) dan mampu menangani beban lebih tinggi (500 request/detik) dibandingkan *framework* lain
15.	Muanzir Yusuf & Muhammad Zaid[18]	2016	Sistem ini menggunakan Arduino Mega 2560 dan modul GSM/GPRS SIM900A untuk mengirim data. Platform *website*-nya dibangun dengan bahasa pemrograman PHP, HTML, CSS, dan jQuery, serta	Waktu yang dibutuhkan untuk mengirim data dari alat ke *database server* adalah sekitar 3-6 detik, tergantung pada kualitas jaringan GSM/GPRS. Sistem *database server* dan *datalogger* lokal pada alat berfungsi dengan baik,

			menggunakan MySQL sebagai database yang dikelola melalui XAMPP dan phpMyAdmin	memastikan tidak ada data yang hilang jika jaringan terputus. *Website* berhasil menampilkan semua data sensor kualitas udara (suhu, kelembapan, tekanan, CO, NO_2, SO_2, O_3, dan PM10) dalam bentuk grafik secara *online*
16.	Nazwa Rafilla & Anggi Rizky Windra Putri[19]	2025	simulator yang dibangun pada platform Wokwi. Mikrokontroler ESP32 digunakan untuk mengirimkan data dari sensor MQ-135 ke platform Antares. Data dikirimkan secara *real-time* melalui protokol HTTP dengan format JSON untuk pemantauan jarak jauh	Hasil pengujian *black box testing* menunjukkan sistem berhasil mengirimkan data dari simulator ke platform Antares dengan sukses Platform Antares terbukti mampu menerima dan menyimpan data sensor secara efektif, yang mencakup nilai PPM dan status kualitas udara ("BAIK" atau "TIDAK BAIK") .Sistem secara keseluruhan mampu berfungsi sesuai harapan untuk pemantauan kualitas udara yang efisien melalui platform Antares.
17.	Firga Deman Samudra, Miftachul ulum, Koko joni, Diana Rahmawati*[20]*	2021	Sistem ini menggunakan Arduino Uno dan NodeMCU untuk mengolah data dari beberapa sensor (MQ-7, MQ-135, MQ-131, Sharp). Data kualitas udara kemudian dikirim ke platform Antares IoT Cloud untuk monitoring. Data pada platform Antares dapat diakses secara *real-*	Platform Antares berhasil menampilkan data dari semua sensor (CO, CO2, Ozon, Debu) dalam bentuk grafik yang dapat diakses melalui *website* dan *smartphone*. • Pengujian : Quality of Service (QoS) untuk pengiriman data ke platform menunjukkan hasil yang cukup bagus. Berdasarkan indeks QoS, performa jaringan mendapatkan kategori

			time melalui antarmuka web dan *smartphone*.	"Sangat Bagus" untuk parameter throughput, delay, dan packet loss.
18.	Roni Stiawan, *et al[21]*	2015	Arduino Mega dan GSM Shield untuk mengirim data sensor cuaca. Platform *website*-nya dibangun menggunakan PHP untuk menerima dan mengolah data , yang kemudian disimpan dalam *database* MySQL. Untuk visualisasi data, *website* menggunakan HTML, JavaScript, dan *library* Flot Charts untuk membuat grafik yang dinamis	Web server berhasil menerima dan menyimpan data cuaca yang dikirim dari perangkat keras ke *database* MySQL. *Website* berhasil menampilkan data cuaca dalam dua bentuk visualisasi: tabel data terbaru dan grafik dinamis. Fitur visualisasi grafik pada *website* memungkinkan pengguna untuk memilih data berdasarkan tanggal yang ingin ditampilkan. Sistem secara keseluruhan terbukti berfungsi dengan baik setelah diuji coba monitoring selama sembilan jam.
19.	Hendi Budianto & Budi Sumanto[22]	2024	Sistem ini menggunakan Arduino Uno untuk mengolah data dan NodeMCU ESP8266 untuk konektivitas internet. Platform website dibangun menggunakan PHP dengan *framework* Bootstrap, serta MySQL sebagai *database*. Data dari perangkat dikirim ke *server* setiap 17 detik	Website berhasil menampilkan data sensor secara *real-time* dalam bentuk angka, grafik, dan tabel.Data pada *website* diperbarui setiap 17 detik .Sistem *database* berhasil menyimpan semua data yang masuk untuk inventarisasi dan pelaporan .Terdapat fitur tambahan pada *website* berupa notifikasi peringatan kualitas udara buruk dan fitur ekspor data (CSV, PDF, xlsx)
20	Camelia Oktaviani, *et al.[23]*	2025	Sistem informasi ini merupakan aplikasi berbasis web yang diakses pada *desktop*. Platform	Semua fitur utama yang diuji (Login, Logout, Tambah Barang, Quality Control, Ekspor Data) dinyatakan "Berhasil" berdasarkan pengujian

			dibangun menggunakan *framework* Laravel dengan *database* MySQL. Aplikasi dijalankan pada *server* lokal menggunakan Laragon. Proses perancangan sistem menggunakan pemodelan UML (*Use Case, Activity, Sequence, Class Diagram*).	fungsional. Sistem terbukti mampu mengoptimalkan pencatatan stok (barang masuk, keluar, rusak) secara otomatis.Secara kualitatif, platform ini berhasil meningkatkan kecepatan pencarian data, mengurangi kesalahan pencatatan manual, dan menghasilkan laporan stok yang lebih akurat dan *real-time*.

1.2 Rumusan Masalah

Adapun rumusan masalah dari Tugas Akhir ini, sebagai berikut:

1. Bagaimana mengembangkan sistem pemantauan kualitas udara berbasis *Internet of Things* (IoT dengan pengiriman data melalui LoRa?

2. Bagaimana merancang integrasi antara perangkat sensor dengan platform Antares dan phpMyAdmin untuk penyimpanan data sementara dan lokal?

3. Bagaimana membangun antarmuka *website* menggunakan Laravel dan Laragon untuk menampilkan data kualitas udara secara *real-time*?

4. Sejauh mana sistem yang dikembangkan mampu berfungsi secara efektif dan akurat dalam memantau serta mengelola data kualitas udara?

1.3 Tujuan

Adapun tujuan dari Tugas Akhir ini, sebagai berikut:

1. Mengembangkan sistem pemantauan kualitas udara berbasis *Internet of Things* (IoT) yang mampu mengukur parameter lingkungan seperti suhu, kelembapan, dan gas berbahaya menggunakan sensor yang terintegrasi.

2. Merancang proses pengiriman data sensor menggunakan protokol komunikasi LoRa ke platform Antares sebagai media penyimpanan sementara.

3. Membangun sistem backend menggunakan Laragon dan phpMyAdmin untuk menyimpan data ke database lokal secara otomatis.

4. Merancang sistem pemantauan kualitas udara berbasis *website* menggunakan Laravel secara *Real time*.

5. Menguji dan mengevaluasi kinerja sistem dalam menampilkan dan mengelola data pemantauan kualitas udara secara akurat, stabil, dan mudah diakses. serta memastikan sistem mudah diakses oleh publik melalui *website* yang di-*hosting* secara daring.

1.4 Cakupan Pengerjaan

Untuk menjaga agar penelitian ini tetap fokus dan terarah, maka pengembangan sistem pemantauan kualitas udara ini dibatasi pada ruang lingkup.

Cakupan pengerjaan berisi:

1. Perancangan sistem monitoring kualiatas udara ini menggunakan struktur frame work Laravel

2. Sistem ini tidak dikembangkan dalam bentuk aplikasi mobile native untuk Android maupun iOS.

3. Pengembangan dan pengujian *website* dilakukan pada lingkungan server lokal menggunakan Laragon, sedangkan untuk implementasi akhir, *website* di-*hosting* secara daring agar dapat diakses oleh publik.

1.5 Tahapan Pengerjaan

Penelitian ini akan dilaksanakan melalui tahapan – tahapan sistematis, sebagai berikut:

1. Studi Literatur

   Tahap awal ini berfokus Melakukan pemahaman untuk pengembangan web dan analisis kebutuhan fungsional. terhadap teori-teori pendukung yang berkaitan dengan kualitas udara,sensor lingkungan ,protokol komunikasi LoRa,platform Antares,serta pengembangan *website* berbasis *framework* Laravel, pengembangan website server lokal Laragon, visualisasi data dan mengIntegrasikan data dari Antares ke database dan *website*

2. Analisis Kebutuhan dan Pengumpulan Data Teknis

   Tahap ini bertujuan untuk mendefinisikan secara rinci spesifikasi dan kebutuhan fungsional dari *website monitoring* yang akan dikembangkan. Analisis difokuskan pada kebutuhan data yang akan diolah, di mana sistem harus mampu menerima dan memproses data dari platform Antares yang mencakup sensor apa saja yang digunakan .
   Kebutuhan perangkat lunak utama yang ditetapkan meliputi framework Laravel untuk pembangunan backend dan frontend, basis data MySQL yang

dikelola melalui phpMyAdmin untuk penyimpanan data historis, dan Laragon sebagai lingkungan server lokal. Secara fungsional, website harus mampu menampilkan data secara *real-time* melalui *gauge* dan data historis dalam bentuk grafik. Adapun kebutuhan non-fungsional yang menjadi prioritas adalah antarmuka pengguna yang responsif, intuitif, dan memiliki waktu muat yang optimal.

3. Perancangan Sistem dan Antarmuka

   Tahap ini mencakup perancangan arsitektur sistem dan alur komunikasi data dari sensor ke antares dan database hingga ke tampilan *website*. Arsitektur sistem digambarkan melalui diagram alir dan diagram arsitektur yang dibuat menggunakan draw.io. Desain antarmuka *website* dirancang menggunakan Figma untuk memastikan kemudahan penggunaan dan keterbacaan visualisasi data. Sistem dirancang agar data yang diterima oleh platform Antares dapat disimpan secara otomatis ke dalam database MySQL dan divisualisasikan melalui Laravel dalam bentuk grafik dan indikator gauge.

4. Implementasi Sistem

   Tahap implementasi berfokus pada realisasi menjadi aplikasi fungsional, dengan server lokal sebagai lingkungan kerja utama.. Proses ini dibagi menjadi dua bagian utama yang berjalan secara paralel, yaitu pengembangan *backend* dan *frontend*. Pada sisi backend, pengembangan dilakukan menggunakan framework Laravel di lingkungan Laragon. Logika utama dibangun untuk menangani proses penarikan data secara periodik dari API platform Antares, yang kemudian data tersebut diolah dan disimpan secara terstruktur ke dalam basis data MySQL. Selain itu, disiapkan juga *endpoint* API internal untuk menyediakan data yang siap dikonsumsi oleh sisi *frontend*. Pada sisi frontend, desain antarmuka yang telah dirancang diimplementasikan menjadi kode HTML dan CSS menggunakan sistem *template* Blade Laravel. Untuk visualisasi data, diintegrasikan *library* grafik berbasis JavaScript. Skrip dikembangkan untuk mengambil data dari API internal *backend* dan menampilkannya secara dinamis pada komponen visual seperti gauge untuk data terkini dan grafik garis untuk data historis. Implementasi dilakukan secara iteratif untuk memastikan setiap fitur berfungsi sesuai dengan perancangan.

5. Pengujian dan Evaluasi Sistem

Tahap akhir pengerjaan dimulai dengan deployment sistem, di mana aplikasi *website* yang telah dikembangkan di lingkungan lokal diunggah dan dikonfigurasi pada server *hosting* daring. Langkah ini bertujuan agar sistem dapat diakses secara publik melalui internet.Setelah sistem berhasil di-*hosting*, proses dilanjutkan dengan pengujian dan evaluasi menyeluruh di lingkungan *live* Proses ini dimulai dengan pengujian sisi backend untuk memvalidasi integrasi data dari API Antares dan fungsionalitas API internal menggunakan perangkat lunak seperti Postman. Pengujian kemudian dilanjutkan pada sisi frontend, yang mencakup pengujian kompatibilitas pada berbagai *web browser* untuk memastikan responsivitas antarmuka serta keakuratan visualisasi data pada *gauge* dan grafik. Terakhir, dilakukan evaluasi sistem secara keseluruhan berdasarkan parameter performa seperti kecepatan muat halaman, validitas data yang ditampilkan, dan *usability* antarmuka, di mana hasil dari evaluasi ini menjadi dasar untuk melakukan penyempurnaan akhir pada *website monitoring*.