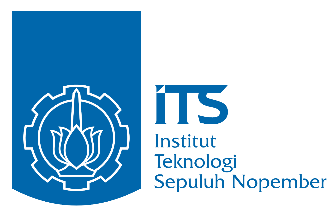
HALAMAN

JUDUL



**INTERKONEKSI SISTEM INSTRUMENTASI-VI231418**

CORNTECH: Sistem Monitoring Suhu dan Kelembaban Terintegrasi IoT dan Blockchain untuk Penyimpanan Aman dan Berkualitas Komoditas Pertanian

**ANDRE MAHESA BAGASKARA (2042231012)**

**CAHYO OKTO RISFIAN (2042231044)**

**SITI AISYAH (2042231062)**

Dosen Pengampu

**Ahmad Radhy, S.Si., M.Si.**

NPP. 1983201711054

**Program Studi Rekayasa Teknologi Instrumentasi**

Departemen Teknik Instrumentasi

Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Surabaya

2025



**INTERKONEKSI SISTEM INSTRUMENTASI-VI231418**

**CORNTECH: Sistem Monitoring Suhu dan Kelembaban Terintegrasi IoT dan Blockchain untuk Penyimpanan Aman dan Berkualitas Komoditas Pertanian**

**ANDRE MAHESA BAGASKARA (2042231012)**

**CAHYO OKTO RISFIAN (2042231044)**

**SITI AISYAH (2042231062)**

Dosen Pengampu

**Ahmad Radhy, S.Si., M.Si.**

NPP. 1983101711054

**Program Studi** **D4 Rekayasa Teknologi Instrumentasi**

Departemen Teknik Instrumentasi

Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Surabaya

2025

**BAB I**

**PENDAHULUAN**

1. **LATAR BELAKANG**

Sektor pertanian merupakan tulang punggung ketahanan pangan nasional sekaligus penopang utama perekonomian di berbagai daerah. Namun, salah satu permasalahan krusial dalam rantai pasok hasil pertanian adalah rendahnya kualitas penyimpanan komoditas akibat keterbatasan sistem monitoring lingkungan yang andal. Faktor suhu dan kelembaban yang tidak terkontrol sering menyebabkan penurunan mutu produk, timbulnya jamur, serta kehilangan hasil pasca panen yang signifikan. Hal ini tidak hanya menurunkan nilai jual produk tetapi juga memperburuk kepercayaan konsumen terhadap kualitas pangan lokal (Kementerian Pertanian RI, 2022).

Seiring berkembangnya teknologi, muncul peluang untuk menghadirkan solusi berbasis Internet of Things (IoT) guna mendukung monitoring presisi kondisi lingkungan penyimpanan. IoT memungkinkan integrasi sensor untuk membaca data suhu dan kelembaban secara otomatis dan real-time, serta mengirimkan data tersebut ke server untuk diolah lebih lanjut. Namun, tantangan berikutnya adalah bagaimana memastikan data yang dikumpulkan benar-benar valid, aman, dan tidak dapat dimanipulasi oleh pihak-pihak tertentu, terutama dalam konteks rantai pasok yang melibatkan banyak pemangku kepentingan (Zhang et al., 2020).

Teknologi blockchain hadir sebagai jawaban atas tantangan tersebut dengan kemampuannya mencatat data secara immutable, transparan, dan terdesentralisasi. Dengan menggabungkan IoT dan blockchain, sistem monitoring tidak hanya sekadar membaca data sensor, tetapi juga mampu mencatat riwayat kondisi lingkungan penyimpanan secara permanen dan dapat diverifikasi kapan saja. Pendekatan ini akan meningkatkan kepercayaan seluruh pihak yang terlibat, mulai dari petani, distributor, hingga konsumen akhir, bahwa produk pertanian disimpan sesuai standar mutu yang ditetapkan (Tan & Lim, 2021; Nakamoto, 2008).

Implementasi sistem monitoring suhu dan kelembaban berbasis IoT dan blockchain sangat relevan untuk memperkuat traceability, transparansi, dan akuntabilitas dalam rantai pasok hasil pertanian Indonesia. Sistem ini menjadi langkah strategis dalam mendukung transformasi digital sektor pertanian, meningkatkan daya saing produk lokal, dan menjawab tuntutan pasar global yang semakin peduli terhadap keamanan serta kualitas pangan. Dengan demikian, pembangunan sistem ini diharapkan menjadi kontribusi nyata dalam mewujudkan rantai pasok pangan yang modern, aman, dan berkelanjutan (FAO, 2021; BPS, 2022).

1. **RUMUSAN MASALAH**

Berdasarkan latar belakang yang telah diuraikan, proyek ini berangkat dari permasalahan utama dalam pengelolaan gudang jagung yang masih belum optimal, khususnya dalam hal pemantauan kondisi lingkungan dan transparansi data. Adapun rumusan masalah yang dapat dirumuskan adalah sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan sistem monitoring suhu dan kelembaban yang andal untuk mendukung pengelolaan kondisi lingkungan gudang jagung agar kualitas komoditas tetap terjaga selama penyimpanan?
2. Bagaimana membangun arsitektur komunikasi berbasis Modbus RTU dan TCP Server (Rust) untuk menjamin akurasi dan kecepatan pertukaran data antara sensor dan pusat pengolahan data?
3. Bagaimana merancang penyimpanan data suhu dan kelembaban secara time-series menggunakan InfluxDB serta menyediakan visualisasi real-time melalui dashboard Grafana yang mudah diakses?
4. Bagaimana mengembangkan aplikasi desktop berbasis Qt untuk memberikan pemantauan lokal dengan fitur kontrol dan indikator status sistem secara interaktif dan user-friendly?
5. Bagaimana mengintegrasikan teknologi blockchain dan Web3 untuk menjamin keutuhan (integritas), transparansi, dan traceability data kondisi lingkungan dalam ekosistem rantai pasok jagung?
6. **TUJUAN PENELITIAN**

Penelitian ini bertujuan untuk merancang dan mengimplementasikan prototipe sistem monitoring suhu dan kelembaban gudang jagung berbasis IoT yang terintegrasi dengan teknologi blockchain dan Web3 untuk mendukung transparansi dan traceability dalam rantai pasok komoditas pertanian. Secara khusus, tujuan yang ingin dicapai adalah:

1. Membangun sistem monitoring suhu dan kelembaban menggunakan sensor SHT20 dan mikrokontroler ESP32-S3 dengan komunikasi Modbus RTU yang dapat bekerja secara andal di lingkungan penyimpanan jagung.
2. Mengimplementasikan arsitektur komunikasi client-server berbasis TCP Server (Rust) untuk menerima dan memproses data sensor secara real-time.
3. Mengembangkan sistem penyimpanan data berbasis time-series menggunakan InfluxDB serta menyediakan visualisasi kondisi lingkungan secara real-time melalui dashboard Grafana.
4. Merancang dan membuat aplikasi desktop berbasis Qt (Qt5/Qt6) untuk monitoring lokal dengan tampilan antarmuka yang informatif dan mudah digunakan.
5. Mengintegrasikan sistem monitoring dengan teknologi blockchain dan Web3 untuk mencatat data suhu dan kelembaban secara immutable, sehingga mendukung transparansi, akuntabilitas, dan keterlacakan (traceability) dalam rantai pasok jagung.
6. **MANFAAT PENELITIAN**
7. Manfaat Akademik
   1. Memberikan kontribusi dalam pengembangan ilmu di bidang interkoneksi sistem instrumentasi, khususnya dalam integrasi antara sistem monitoring lingkungan, database time-series, dan teknologi blockchain.
   2. Menjadi contoh penerapan nyata konsep transparansi data dan traceability dalam rantai pasok berbasis teknologi Web3, yang dapat menjadi referensi untuk penelitian sejenis di bidang pertanian dan pangan.
   3. Meningkatkan pemahaman mahasiswa terhadap penggunaan protokol komunikasi industri (seperti Modbus RTU/TCP), pengolahan data sensor, dan integrasi dengan sistem visualisasi serta smart contract.
8. Manfaat Praktis
   1. Menyediakan solusi sistematis untuk monitoring suhu dan kelembaban gudang jagung secara real-time tanpa ketergantungan pada perangkat mikrokontroler, memanfaatkan infrastruktur industri yang telah ada.
   2. Meningkatkan kualitas penyimpanan jagung dengan menghadirkan sistem peringatan dini (early warning) dan dokumentasi lingkungan yang dapat digunakan sebagai bukti kepatuhan terhadap standar penyimpanan pangan.
   3. Mendukung kepercayaan antar pihak dalam rantai pasok, seperti petani, distributor, dan pemerintah, melalui pencatatan hash data ke blockchain yang tidak dapat dimanipulasi dan dapat diaudit publik.
9. Manfaat Sosial dan Ekonomi
   1. Membantu mengurangi kerugian pascapanen akibat penyimpanan yang tidak sesuai standar, sehingga berdampak pada peningkatan daya saing petani dan UMKM pangan.
   2. Menunjang program pemerintah dalam transformasi digital sektor pertanian dan pangan, sekaligus membuka peluang pengembangan model bisnis berbasis transparansi dan akuntabilitas data.
   3. Mendorong penggunaan teknologi decentralized ledger dan open-source monitoring di sektor-sektor kritis, sebagai bagian dari adopsi Industri 4.0 dan Agriculture 4.0 di Indonesia.
10. **BATASAN PENELITIAN**
11. Sistem hanya menggunakan sensor industri yang mendukung protokol Modbus RTU atau Modbus TCP/IP, yang terhubung langsung ke komputer server atau gateway. Proyek ini tidak menggunakan mikrokontroler (seperti ESP32, Arduino, atau STM32) dalam tahap akuisisi data.
12. Parameter lingkungan yang dimonitor terbatas pada suhu dan kelembaban relatif udara di dalam ruang penyimpanan gudang jagung. Parameter lain seperti kadar gas, tekanan, atau intensitas cahaya tidak termasuk dalam cakupan sistem ini.
13. Sistem hanya diterapkan dan diuji coba pada simulasi ruang penyimpanan berskala kecil atau skenario gudang jagung tunggal. Pengujian multi-lokasi atau integrasi antar gudang belum menjadi fokus penelitian ini.
14. Visualisasi data dilakukan melalui dashboard Grafana yang terhubung ke database InfluxDB. Sistem tidak menyediakan fitur mobile app atau dashboard berbasis cloud dalam lingkup proyek ini.
15. Integrasi dengan blockchain dibatasi pada pencatatan hash data suhu dan kelembaban dalam interval waktu tertentu (misalnya setiap 6 jam) ke dalam smart contract berbasis Ethereum. Proyek ini tidak mencatat seluruh data sensor ke blockchain secara langsung, guna menghindari beban transaksi dan biaya gas yang tinggi.

**BAB II**

**TINJAUAN PUSTAKA**

1. ***State of Art***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Judul Penelitian** | **Penulis (Tahun)** | **Metode yang Digunakan** | **Hasil yang Didapatkan** |
| 1 | Analisis Kualitas Jagung Berbasis IoT dengan Penerapan Model SSD Mobilenet dan Histogram | Audy dan Zaini (2022) | Metode rekayasa sistem: Penelitian ini merancang, membangun, dan mengimplementasikan suatu sistem monitoring pertumbuhan jagung dan sistem identifikasi kualitas jagung. | Sistem berbasis IoT memudahkan petani untuk memantau kondisi lahan tanpa harus hadir langsung di lokasi. |
| 2 | Sistem Monitoring pH Tanah, Suhu dan Kelembaban Tanah pada Tanaman Jagung Berbasis Internet of Things (IoT) | Brigida Helvia Vien, Ferry Hadary, Erlinda Yurisinthae (2023) | Metode eksperimental: Penelitian ini menggunakan metode eksperimental karena melibatkan perancangan, pembuatan, dan pengujian sistem monitoring secara langsung di lapangan. | Sistem mampu memonitor pH tanah, suhu, dan kelembaban tanah dengan baik menggunakan sensor pH tanah, sensor DS18B20, dan sensor YL-69. |
| 3 | Towards an IoT and Blockchain-based System for Monitoring and Tracking Agricultural Products | Tuyen Phong Truong, Minh Chau Phan, dan Huy Quang Nguyen (2023) | Penelitian ini fokus pada perancangan, pembangunan, integrasi, dan implementasi sistem monitoring dan traceability produk pertanian berbasis IoT dan blockchain. | Sistem berhasil memonitor kondisi penyimpanan dan transportasi produk pertanian (suhu, kelembaban, lokasi) secara real-time, dan mencatat data ke blockchain untuk menjamin transparansi dan integritas. |
| 4 | A Paradigm Shift in Agriculture: Integrating IoT and Blockchain for Automated Harvesting System | Kunchapu Devakumar, Degala Saranya, Degala Jwalitha, Indla Sneha Latha (2024) | Penelitian ini menggunakan metode rekayasa sistem karena berfokus pada perancangan, pembangunan, dan integrasi sistem otomatisasi panen berbasis IoT dan blockchain. | Sistem ini menghasilkan data real-time dari sensor untuk mendukung precision agriculture. |

Berdasarkan kajian literatur sebelumnya, telah banyak penelitian yang mengembangkan sistem monitoring suhu, kelembaban, dan kondisi lingkungan berbasis Internet of Things (IoT), baik dalam konteks gudang pertanian, rumah kaca, lahan budidaya, maupun transportasi produk pertanian. Sebagian besar sistem tersebut menggunakan metode pemantauan berbasis mikrokontroler (seperti ESP32, ESP8266, Raspberry Pi, atau Arduino) yang mengirimkan data ke cloud server (misalnya ThingSpeak, ANTARES) atau menyediakan visualisasi melalui dashboard berbasis web atau SCADA lokal. Selain itu, sebagian besar penelitian lebih menitikberatkan pada akuisisi data dan visualisasi real-time, tanpa mengintegrasikan mekanisme yang kuat untuk menjamin integritas data jangka panjang, transparansi lintas rantai pasok, serta auditabilitas bagi berbagai pemangku kepentingan.

Salah satu kelemahan umum yang teridentifikasi adalah tidak adanya mekanisme untuk menjamin bahwa data suhu, kelembaban, dan kondisi lainnya tidak dimanipulasi setelah dikirim dari sensor. Data sering kali hanya tersimpan di cloud atau lokal tanpa sistem yang memastikan integritas dan keasliannya secara end-to-end. Beberapa penelitian, seperti oleh Truong et al. (2023) dan Kunchapu et al. (2024), telah mencoba menggabungkan IoT dengan blockchain untuk menjamin traceability dan keamanan data pada proses transportasi atau sistem panen, namun implementasi tersebut lebih terfokus pada aspek supply chain atau harvesting system dan belum secara spesifik mengadopsi arsitektur time-series database dengan visualisasi dual-mode (misalnya Grafana + aplikasi desktop lokal berbasis Qt). Selain itu, hingga saat ini belum ditemukan literatur yang secara spesifik mengembangkan sistem monitoring suhu dan kelembaban untuk gudang jagung sektor strategis dalam rantai pasok pangan di negara berkembang dengan fitur integrasi penyimpanan time-series (InfluxDB), visualisasi dual-mode (Grafana dan Qt GUI), serta pencatatan hash summary data ke dalam smart contract berbasis Ethereum sebagai upaya memperkuat aspek transparansi, auditabilitas, dan keandalan data lingkungan gudang jagung.

1. **Pergudangan Jagung sebagai Plant Monitoring**

Pergudangan jagung merupakan salah satu komponen penting dalam rantai pasok pangan nasional yang berfungsi sebagai tempat penyimpanan hasil panen sebelum didistribusikan ke pasar, distributor, atau konsumen akhir. Kualitas jagung selama masa penyimpanan sangat dipengaruhi oleh kondisi lingkungan di dalam gudang, terutama suhu dan kelembaban, yang dapat menyebabkan penurunan mutu fisik, biologis, dan nutrisi apabila tidak dikendalikan secara tepat (FAO, 2019). Brigida et al. (2023) menekankan pentingnya pengendalian suhu dan kelembaban dalam budidaya jagung untuk mencegah penurunan kualitas akibat faktor lingkungan, namun implementasi monitoring suhu dan kelembaban secara khusus pada gudang jagung masih jarang dibahas dalam literatur. Dalam konteks sistem instrumentasi, gudang penyimpanan jagung diposisikan sebagai *plant* yakni objek fisik yang menjadi fokus pemantauan dan pengendalian sistem. Sebagai *plant*, gudang memiliki karakteristik lingkungan tertutup, berskala luas, dan menyimpan komoditas pangan dalam jumlah besar yang sensitif terhadap fluktuasi mikroklimat. Oleh karena itu, diperlukan sistem monitoring yang mampu melakukan pengukuran secara berkelanjutan, real-time, dan terintegrasi. Sejumlah penelitian seperti Truong et al. (2023) dan Kunchapu et al. (2024) telah menggabungkan IoT dengan blockchain untuk monitoring pada konteks transportasi dan sistem panen, tetapi belum fokus pada monitoring suhu dan kelembaban di fasilitas penyimpanan jagung.

Menurut Suliantoro et al. (2021), kelembaban di atas 70% dapat memicu pertumbuhan jamur seperti *Aspergillus flavus* yang menghasilkan aflatoksin, sementara suhu di atas 30°C dapat mempercepat aktivitas hama gudang seperti *Sitophilus zeamais*. Tanpa pemantauan yang tepat, kondisi ini dapat dengan cepat menurunkan kualitas jagung dan menyebabkan kerugian ekonomi. Dalam proyek ini, gudang jagung tidak hanya diposisikan sebagai ruang penyimpanan pasif, melainkan sebagai entitas aktif dalam ekosistem digital. Dengan mengadopsi teknologi IoT untuk pengukuran suhu dan kelembaban, sistem ini juga terintegrasi dengan blockchain (Web3) untuk mencatat hash data secara periodik ke dalam smart contract berbasis Ethereum. Hal ini mendukung transparansi, traceability, dan auditabilitas lintas pemangku kepentingan (Truong et al., 2023; Kunchapu et al., 2024). Integrasi dengan database time-series (InfluxDB) dan visualisasi dual-mode (Grafana + Qt GUI) menjadi kontribusi baru yang belum dijumpai pada literatur sebelumnya terkait monitoring gudang jagung.

1. **Sistem Monitoring Lingkungan**

Sistem monitoring lingkungan merupakan sistem terintegrasi yang bertujuan untuk mengamati, mencatat, dan menganalisis parameter lingkungan secara berkala dan real-time. Dalam konteks penyimpanan beras, parameter yang paling berpengaruh terhadap stabilitas mutu adalah suhu dan kelembaban relatif udara. Kedua variabel ini berperan langsung dalam mencegah pertumbuhan mikroorganisme, serangan hama, serta proses degradasi yang dapat menyebabkan kehilangan pascapanen (*postharvest loss*). Berdasarkan laporan FAO (2019), suhu gudang yang ideal untuk penyimpanan beras berkisar antara 15–25°C, sedangkan kelembaban relatif harus dijaga di bawah 70%. Jika kelembaban melebihi ambang tersebut, kondisi lingkungan menjadi optimal untuk pertumbuhan jamur seperti *Aspergillus flavus*, yang menghasilkan aflatoksin berbahaya bagi kesehatan manusia (Tian, 2016).

Oleh karena itu, sistem monitoring lingkungan yang presisi diperlukan agar pengelola gudang dapat mengambil tindakan korektif lebih awal, seperti menyalakan sistem ventilasi, pengering udara, atau melakukan pemindahan stok. Sistem ini juga dapat berfungsi sebagai bukti dokumentasi digital bahwa kondisi penyimpanan telah sesuai standar, terutama jika terintegrasi dengan blockchain. Pada proyek ini, sistem menggunakan **sensor industri tipe SHT20** atau sensor sejenis yang mendukung **protokol komunikasi Modbus RTU**. Sensor ini berfungsi sebagai komponen utama dalam proses akuisisi data lingkungan. SHT20 merupakan sensor digital yang menggabungkan **pengukuran suhu dan kelembaban** dalam satu chip dan dirancang untuk lingkungan industri dengan **akurasi tinggi** (±0.3°C untuk suhu dan ±3% untuk RH). Menurut Yasin et al. (2020), penggunaan sensor digital seperti SHT20 atau DHT22 dalam sistem monitoring pangan memberikan keunggulan dalam hal **presisi, daya tahan terhadap lingkungan ekstrem, serta kemampuan integrasi ke sistem SCADA atau IoT**. Sensor ini sangat cocok digunakan untuk kebutuhan industri skala kecil maupun besar karena memiliki **kalibrasi pabrik dan protokol standar industri.**

1. **Komunikasi Data Industri**

Komunikasi data industri merupakan sistem pertukaran informasi antar perangkat di lingkungan otomasi dan kontrol industri. Dalam konteks sistem monitoring lingkungan, komunikasi ini menghubungkan sensor industri dengan sistem pusat seperti server, PLC, atau HMI agar data dapat diolah, disimpan, dan divisualisasikan. Keandalan dan standar protokol komunikasi yang digunakan sangat menentukan kualitas dan akurasi sistem secara keseluruhan (Palau et al., 2021). Modbus RTU adalah protokol komunikasi serial berbasis RS-485 yang banyak digunakan dalam industri untuk menghubungkan perangkat seperti sensor, aktuator, dan kontroler dalam topologi *master-slave*. Dalam protokol ini, sensor bertindak sebagai slave yang merespons permintaan dari master (komputer/server atau gateway), dan pertukaran data terjadi dalam format biner yang efisien.

Menurut Kumar et al. (2020), Modbus RTU tetap menjadi standar komunikasi penting dalam sistem monitoring karena konsumsi bandwidth-nya rendah, cocok untuk implementasi real-time, dan mudah dikonversi ke protokol lain menggunakan converter/bridge. Modbus TCP/IP adalah versi berbasis jaringan Ethernet dari protokol Modbus. Protokol ini bekerja dengan mengemas data Modbus RTU ke dalam format TCP/IP dan memungkinkan pengiriman data melalui jaringan LAN atau WAN. Ini sangat cocok untuk sistem modern yang terhubung ke jaringan komputer atau cloud. Menurut Palau et al. (2021), Modbus TCP/IP sangat sesuai untuk sistem instrumentasi cerdas karena mudah diintegrasikan dengan *software stack* modern dan memiliki skalabilitas tinggi untuk sistem multisensor. Dalam proyek ini, jika sensor atau gateway sudah mendukung komunikasi Ethernet, maka Modbus TCP/IP digunakan untuk menghubungkan sensor ke program TCP Server yang berjalan di komputer. Program ini kemudian menyimpan data ke InfluxDB, dan mencatat *hash* ke blockchain.

1. **InfluxDB**

InfluxDB adalah database time-series (TSDB) open-source yang dirancang khusus untuk menangani data yang berkaitan dengan waktu (*timestamped data*), seperti suhu, kelembaban, tegangan, dan tekanan. Berbeda dengan database relasional (SQL), InfluxDB dioptimalkan untuk kecepatan penulisan (*write performance*), efisiensi penyimpanan data sensor secara kontinu, serta analisis data berbasis waktu secara real-time (Barksdale & Stone, 2019). Dalam sistem monitoring lingkungan, InfluxDB berperan penting sebagai media penyimpanan data suhu dan kelembaban yang diperoleh dari sensor. Setiap data dicatat bersama dengan waktu pengambilan data, sehingga memungkinkan pengguna untuk menganalisis perubahan kondisi lingkungan secara historis maupun real-time.

InfluxDB juga mendukung pengaturan retention policy, yaitu kebijakan penyimpanan data agar otomatis terhapus setelah waktu tertentu, dan continuous query yang dapat digunakan untuk membuat ringkasan data seperti rata-rata suhu per 6 jam. Ringkasan ini bermanfaat sebagai sumber hash yang akan dicatat ke blockchain untuk menjamin integritas data. Menurut Srinivasan et al. (2022), arsitektur InfluxDB sangat cocok digunakan dalam sistem monitoring industri karena mampu menyimpan jutaan entri sensor dengan efisien dan tetap mendukung integrasi dengan platform visualisasi dan blockchain.

1. **Visualisasi Data dengan Grafana**

Grafana adalah sebuah platform open-source untuk visualisasi dan analisis data yang dirancang khusus untuk data time-series. Dengan Grafana, pengguna dapat membuat dashboard interaktif, menyusun grafik, dan memantau data sensor secara real-time dari berbagai sumber seperti InfluxDB, Prometheus, MySQL, atau API eksternal (Gupta & Singh, 2020). Dalam sistem monitoring, visualisasi berperan penting dalam membantu pengguna memahami pola data dan membuat keputusan secara cepat. Grafana memungkinkan penyajian informasi yang kompleks menjadi bentuk grafik yang mudah dipahami melalui widget seperti line chart, bar graph, gauge, table, dan alert panel.

Menurut Gupta dan Singh (2020), Grafana menjadi pilihan utama dalam proyek-proyek pemantauan berbasis IoT karena antarmukanya yang mudah digunakan, dukungan terhadap banyak jenis data, serta kemampuannya dalam memberikan notifikasi otomatis yang kritis terhadap sistem yang dimonitor. Babu & Rao (2022) menunjukkan bahwa sistem pemantauan suhu dan kelembaban berbasis InfluxDB dan Grafana memiliki akurasi visual tinggi dan dapat diimplementasikan pada berbagai sektor penyimpanan, termasuk logistik dan pertanian. Sementara itu, Singh et al. (2021) menegaskan bahwa Grafana sangat cocok digunakan untuk smart agriculture karena kemampuannya dalam mengelola data yang fluktuatif dan dinamis seperti suhu gudang.

1. **Qt sebagai Antarmuka Monitoring Lokal**

Qt Python (menggunakan PyQt5, PyQt6, atau PySide2/6 tanpa Qt Designer) memungkinkan pengembangan aplikasi desktop berbasis GUI secara murni melalui kode, sehingga antarmuka dan logika dapat sepenuhnya dikendalikan melalui skrip Python. Aplikasi ini dirancang menggunakan Qt versi 5 atau lebih baru untuk menampilkan data suhu dan kelembaban secara real-time, sekaligus menyediakan tombol Start dan Stop untuk mengendalikan proses pembacaan sensor. Tanpa menggunakan Qt Designer, semua elemen GUI seperti tombol, label, grafik, dan indikator status dibangun secara manual melalui kode Python, sehingga memudahkan integrasi langsung dengan backend data dan memungkinkan struktur kode yang lebih fleksibel dan modular. Fitur utama aplikasi ini mencakup tampilan suhu dan kelembaban secara real-time yang diperbarui setiap kali data diterima dari TCP Server, serta indikator status koneksi ke sensor dan server. Indikator ini dapat berupa label atau ikon berwarna (misalnya hijau untuk terkoneksi, merah untuk terputus) yang diatur melalui event koneksi dan pemutusan. Tombol Start dan Stop akan mengendalikan inisiasi atau penghentian thread atau event loop untuk membaca data sensor, sementara alur komunikasi TCP diatur agar tidak menghambat antarmuka dengan menggunakan pemrograman berbasis event atau thread terpisah. Semua komponen GUI dirancang melalui struktur layout seperti QVBoxLayout, QHBoxLayout, dan QGridLayout, sehingga aplikasi tetap rapi dan responsif meski dibuat tanpa bantuan designer visual.

Selain fitur utama, aplikasi ini juga dapat diperluas dengan opsional konfigurasi batas suhu/kelembaban di mana pengguna dapat memasukkan nilai ambang melalui input field di GUI. Sistem akan memantau data dan memicu alarm lokal (misalnya melalui dialog pop-up atau indikator visual khusus) jika nilai sensor melampaui batas yang ditetapkan. Pendekatan tanpa Qt Designer memberikan keuntungan karena seluruh UI dan logika terintegrasi erat dalam satu kode Python, memudahkan pengelolaan versi dan otomatisasi proses build untuk berbagai sistem operasi. Dengan desain ini, aplikasi sangat cocok digunakan untuk pemantauan internal gudang atau laboratorium penyimpanan yang memerlukan kontrol penuh atas fungsi dan antarmuka.

1. **Teknologi Blockchain dalam Rantai Pasok**

Blockchain adalah teknologi penyimpanan data terdistribusi yang memungkinkan setiap data tercatat secara immutabel, terdesentralisasi, dan diverifikasi secara independen oleh banyak pihak. Teknologi ini menggunakan ledger digital bersama, di mana setiap blok berisi informasi yang terhubung kriptografis ke blok sebelumnya, membentuk rekam jejak yang tidak dapat dimanipulasi (Tian, 2016). Blockchain kini tidak hanya digunakan untuk kripto seperti Bitcoin dan Ethereum, tetapi juga meluas ke logistik, pertanian, keuangan, dan kesehatan untuk mendukung transparansi, akuntabilitas, dan traceability dalam rantai pasok (Casino et al., 2019). Karakteristik utamanya meliputi sistem decentralized tanpa otoritas tunggal, data yang immutable, transparansi akses, keamanan kriptografis, dan auditability yang memungkinkan pelacakan setiap proses dengan bukti sah.

Dalam rantai pasok pangan seperti beras, kualitas produk sangat dipengaruhi suhu dan kelembaban penyimpanan, namun sistem monitoring konvensional sering tidak menyediakan bukti otentik pemenuhan standar. Blockchain memastikan integritas dan keaslian data sehingga ideal untuk pelacakan mutu dan dokumentasi kondisi lingkungan selama penyimpanan atau distribusi (Lin et al., 2020). Traceability berbasis blockchain mendukung keamanan pangan, mutu, dan efisiensi logistik (Tian, 2016). Selain itu, penggunaan Qt dalam sistem SCADA dan monitoring industri terbukti efektif menyediakan antarmuka yang stabil dan mudah digunakan untuk mendukung operasi lapangan (Altmeyer & Konietzko, 2018).

1. **Decentralized Application (DApp) dan Web3**

Decentralized Application (DApp) adalah aplikasi digital yang berjalan di atas jaringan blockchain publik tanpa server pusat, dengan smart contract sebagai logika otomatis yang dapat diverifikasi semua pihak (Zheng et al., 2020). DApp memiliki frontend seperti aplikasi web biasa, namun backend-nya dikendalikan kode terbuka di blockchain, bukan perusahaan terpusat. Web3 merupakan konsep generasi ketiga teknologi web yang memungkinkan interaksi langsung pengguna dengan blockchain melalui protokol seperti Web3.js atau Ethers.js, memberi kendali penuh atas data dan aset digital untuk mendukung transparansi, keamanan, dan desentralisasi (Buterin, 2020).

Dalam proyek monitoring suhu dan kelembaban gudang jagung ini, DApp dan Web3 digunakan sebagai antarmuka verifikasi publik bagi distributor, pengelola, dan konsumen untuk melihat hash data lingkungan, mengecek integritas data pada waktu tertentu, serta memastikan produk disimpan sesuai standar. DApp dapat diakses melalui browser dengan wallet seperti MetaMask untuk menampilkan log hash dan timestamp sebagai bukti penyimpanan yang aman tanpa membuka data sensor mentah. DApp membantu membangun kepercayaan tanpa pihak ketiga, penting dalam rantai pasok global, termasuk untuk membangun kepercayaan langsung antara produsen dan konsumen akhir (Xie et al., 2019).

**BAB III**

**METODOLOGI PENELITIAN**

1. **Metode Penelitian**

CORNTECH: Sistem Monitoring Suhu dan Kelembaban Terintegrasi IoT dan Blockchain untuk Penyimpanan Aman dan Berkualitas Komoditas Pertanian dirancang menggunakan pendekatan *applied research* dengan metode perancangan prototipe yang bertujuan menghadirkan solusi teknologi nyata untuk monitoring lingkungan gudang penyimpanan. Penelitian ini fokus membangun sistem yang memantau suhu dan kelembaban secara real-time, menyimpan data dengan efisien, menampilkannya melalui dashboard visual, dan memastikan transparansi data melalui pencatatan hash ke blockchain. Pengembangan sistem dilakukan berbasis *System Development Life Cycle* (SDLC) model prototyping, sehingga memungkinkan iterasi cepat, evaluasi fungsional, dan validasi kebutuhan pengguna. Alur pengerjaan CORNTECH terdiri dari beberapa tahap utama, dimulai dengan studi literatur dan analisis kebutuhan untuk merumuskan spesifikasi sistem berbasis referensi akademik dan standar industri penyimpanan pangan. Selanjutnya, dilakukan perancangan arsitektur sistem mencakup aliran data dari sensor Modbus ke TCP Server, penyimpanan di InfluxDB, hingga pencatatan hash ke blockchain dengan teknologi Web3. Implementasi sistem monitoring dilakukan dengan menghubungkan sensor industri ke TCP Server untuk mengirim data ke database, diikuti pembangunan dashboard Grafana yang menampilkan data suhu dan kelembaban secara real-time serta historis, lengkap dengan fitur pemantauan ambang batas.

Tahap berikutnya adalah integrasi blockchain melalui pengembangan DApp yang memungkinkan publik memverifikasi keaslian data berdasarkan hash yang dicatat pada smart contract di jaringan Ethereum. Sistem kemudian diuji untuk memastikan fungsi tiap modul, mulai dari akurasi visualisasi hingga keandalan pencatatan hash dan antarmuka DApp. Seluruh proses dan hasilnya didokumentasikan dalam laporan proyek sebagai dasar analisis keberhasilan sistem dan rekomendasi pengembangan untuk skala lebih besar atau multi-lokasi, mendukung transformasi digital sektor pertanian dengan monitoring yang aman, transparan, dan akuntabel.

1. **Alat dan Bahan**

Dalam implementasi sistem monitoring suhu dan kelembaban untuk gudang jagung ini, digunakan kombinasi perangkat keras dan perangkat lunak yang membentuk satu kesatuan sistem monitoring real-time berbasis jaringan dan blockchain. Setiap alat dan bahan dipilih berdasarkan fungsinya dalam mendukung akuisisi data, penyimpanan, visualisasi, serta transparansi informasi.

1. Perangkat Keras

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Alat** | **Spesifikasi / Fungsi** |
| 1 | **Sensor Suhu & Kelembaban** | Sensor industri tipe SHT20 atau sejenis yang mendukung protokol Modbus RTU. Digunakan untuk membaca kondisi lingkungan gudang secara kontinu. |
| 2 | **Converter RS485 to USB** | Digunakan untuk menghubungkan sensor Modbus RTU ke komputer server melalui port USB. |
| 3 | **Komputer Server / Laptop** | Minimal prosesor Intel Core i3, RAM 4 GB. Digunakan sebagai host untuk aplikasi monitoring, database, server TCP, visualisasi, dan aplikasi blockchain. |
| 4 | **Jaringan LAN / Internet (opsional)** | Digunakan jika komunikasi menggunakan Modbus TCP/IP atau untuk koneksi ke jaringan Ethereum saat mencatat hash ke blockchain. |
| 5 | **Power Supply Sensor** | Menyediakan suplai daya stabil untuk sensor agar dapat beroperasi secara terus-menerus. |

1. Perangkat Lunak

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Perangkat Lunak** | **Deskripsi / Fungsi** |
| 1 | **Rust Programming Language** | Digunakan untuk membuat program TCP Server yang menerima data dari sensor dan menyimpannya ke InfluxDB. |
| 2 | **InfluxDB** | Database time-series untuk menyimpan data suhu dan kelembaban berdasarkan timestamp. |
| 3 | **Grafana** | Visualisasi berbasis web yang menampilkan data real-time dan historis dari InfluxDB dalam bentuk grafik dan indikator. |
| 4 | **Qt (Qt5/Qt6)** | Framework GUI untuk membangun **aplikasi desktop lokal** guna memantau suhu dan kelembaban secara real-time, tanpa koneksi internet. |
| 5 | **MetaMask** | Wallet digital untuk mengelola akun Ethereum dan berinteraksi dengan DApp dari browser. |
| 6 | **Web3.js** | Library JavaScript untuk menghubungkan frontend DApp ke smart contract di blockchain. |
| 7 | **Node.js** | Environment untuk menjalankan aplikasi DApp dan server lokal frontend. |
| 8 | **Visual Studio Code** | Digunakan sebagai IDE untuk menulis kode Rust, Solidity, Phyton, dan Aplikasi GUI Qt. |
| 9 | **GitHub** | Untuk manajemen versi kode dan kolaborasi pengembangan tim. |

Dengan dukungan alat dan bahan di atas, sistem dapat dibangun secara end-to-end mulai dari pengambilan data sensor, penyimpanan dan visualisasi lokal menggunakan Qt, pelaporan berbasis web melalui Grafana, hingga pencatatan bukti data ke blockchain. Penggunaan Qt dalam sistem ini melengkapi kebutuhan pengawasan di lingkungan lokal yang tidak selalu terkoneksi internet, sedangkan integrasi Web3 mendukung transparansi publik dalam rantai pasok jagung.

1. **Arsitektur Sistem**

Arsitektur sistem merupakan rancangan struktural dari seluruh komponen yang terlibat dalam proyek, mulai dari perangkat input (sensor), perangkat pemrosesan data, sistem penyimpanan, hingga antarmuka pengguna dan integrasi blockchain. Arsitektur ini dirancang secara modular dan terintegrasi, sehingga memungkinkan sistem bekerja secara real-time, fleksibel, dan transparan.

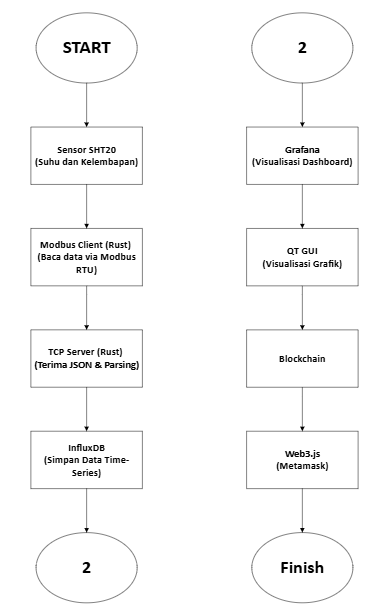
Tujuan dari Arsitektur Sistem:

1. Memastikan data suhu dan kelembaban dapat diakuisisi, disimpan, ditampilkan, dan diverifikasi.
2. Menghubungkan sistem monitoring lokal (Qt) dengan sistem visualisasi web (Grafana).
3. Menjamin integritas data melalui pencatatan hash ke blockchain Ethereum.

Diagram Arsitektur Sistem

A diagram of a computer server

AI-generated content may be incorrect.



Sistem ini terdiri atas beberapa komponen utama sebagai berikut:

1. Sensor Suhu dan Kelembaban (SHT20 / Modbus RTU)

Bertugas mengukur parameter lingkungan (suhu dan kelembaban) di dalam Gudang jagung secara kontinu.

1. Converter RS485 to USB

Menghubungkan sensor berbasis Modbus RTU ke komputer server untuk memungkinkan komunikasi data.

1. Program TCP Server (Rust)

Aplikasi yang dijalankan di komputer server untuk membaca data dari sensor, memprosesnya, dan menyimpan ke database InfluxDB.

1. Database Time-Series (InfluxDB)

Menyimpan data lingkungan secara historis berdasarkan timestamp dengan efisiensi tinggi.

1. Dashboard Web (Grafana)

Menyediakan tampilan visual real-time dan historis yang dapat diakses melalui browser untuk keperluan pemantauan.

1. Aplikasi Monitoring Lokal (Qt5/Qt6)

Menyediakan antarmuka desktop untuk pengguna yang membutuhkan pemantauan tanpa koneksi internet. Tampilan meliputi data suhu dan kelembaban terkini dalam bentuk tabel, grafik, atau indikator.

1. Smart Contract (Solidity)

Kontrak digital yang ditanamkan di blockchain Ethereum untuk menyimpan hash ringkasan data pada interval waktu tertentu (misalnya setiap 6 jam).

1. DApp (Web3.js / Ethers.js)

Aplikasi berbasis web terdesentralisasi yang memungkinkan pengguna memverifikasi hash data dari blockchain dan memastikan data belum dimanipulasi.

1. **Alur Sistem**

CORNTECH merupakan sistem cerdas yang dirancang untuk memantau kondisi lingkungan dalam gudang penyimpanan beras secara real-time dan transparan. Sistem ini diawali dengan pemantauan suhu dan kelembaban oleh sensor industri SHT20 yang terpasang di dalam gudang. Sensor ini bekerja dengan protokol Modbus RTU melalui koneksi RS485, kemudian dihubungkan ke komputer menggunakan konverter RS485-to-USB. Data yang diperoleh diterima oleh program TCP Server berbasis bahasa Rust yang menjalankan proses pembacaan serial, parsing, validasi, hingga penyimpanan data ke dalam InfluxDB sebagai basis data time-series. InfluxDB berperan penting dalam mencatat data berdasarkan waktu untuk keperluan analisis dan visualisasi. Antarmuka pengguna dalam sistem ini dibagi menjadi dua saluran. Pertama adalah aplikasi desktop lokal berbasis Qt yang memberikan visualisasi data melalui grafik dan indikator bagi pengelola gudang. Kedua adalah dashboard Grafana berbasis web yang terhubung langsung ke InfluxDB, memungkinkan pemantauan suhu dan kelembaban secara daring dan historis. Dengan dua jalur ini, pengguna internal dan eksternal dapat mengakses informasi lingkungan gudang secara akurat dan efisien. Selain itu, sistem dilengkapi fitur peringkasan data berkala yang menghasilkan ringkasan metrik penting seperti rata-rata, maksimum, dan minimum.

Untuk menjamin integritas dan keaslian data, sistem CORNTECH melakukan proses hashing pada ringkasan data tersebut menggunakan algoritma seperti SHA-256. Nilai hash yang dihasilkan dicatat ke dalam smart contract di jaringan Ethereum, sehingga menjadi bukti tak terbantahkan atas kondisi lingkungan dalam periode waktu tertentu. Pengguna eksternal seperti distributor atau konsumen dapat memverifikasi data melalui Decentralized Application (DApp) berbasis Web3.js dan MetaMask. Dengan pendekatan ini, CORNTECH tidak hanya menjawab kebutuhan fungsional monitoring gudang, tetapi juga memperkuat prinsip traceability dan akuntabilitas dalam rantai pasok pangan berbasis teknologi Web3.

1. **Implementasi dan Kode Program**

**Semua Program: https://github.com/Risfian111/Interkoneksi-Sistem-Instrumentasi.git**

* 1. **Kode Rust Modbus Client**

Untuk mengambil data dari sensor menggunakan protokol Modbus RTU, langkah pertama adalah menghubungkan sensor ke sistem melalui koneksi serial menggunakan antarmuka RS-485 ke USB. Di sistem operasi Linux, koneksi ini umumnya dikenali sebagai port serial dengan path /dev/ttyUSB0. Sensor seperti SHT20 biasanya memiliki alamat slave 0x01 dalam jaringan Modbus. Selanjutnya, pembacaan data dilakukan dengan mengakses register tertentu pada sensor, seperti alamat register 0x0000 untuk data suhu dan 0x0001 untuk kelembaban. Nilai yang diperoleh dari sensor biasanya berbentuk integer dan perlu dikonversi menjadi float dengan membaginya dengan angka 10, karena sensor mengirimkan data dalam format nilai dikali 10. Setelah data suhu dan kelembaban terbaca dan dikonversi, data tersebut diubah menjadi format JSON untuk memudahkan pengolahan dan transmisi. Data JSON ini kemudian dikirim ke server melalui koneksi TCP menggunakan port 7878 setiap 10 detik, sehingga server menerima pembaruan data sensor secara real-time.

Kode Rust Modbus Client

Main.rs

|  |
| --- |
| use chrono::{Local, SecondsFormat}; use tokio\_modbus::{client::rtu, prelude::\*}; use tokio\_serial::SerialStream; use tokio::{  net::TcpStream,  time::{sleep, Duration},  io::{AsyncReadExt, AsyncWriteExt}, }; use serde\_json::json; use std::error::Error;  async fn sht20(slave: u8) -> Result<Vec<u16>, Box<dyn std::error::Error>> {  let port\_settings = tokio\_serial::new("/dev/ttyUSB0", 9600)  .parity(tokio\_serial::Parity::None)  .stop\_bits(tokio\_serial::StopBits::One)  .data\_bits(tokio\_serial::DataBits::Eight)  .timeout(Duration::from\_secs(1));    let port = SerialStream::open(&port\_settings)?;  let slave = Slave(slave);  let mut ctx = rtu::attach\_slave(port, slave);  let response = ctx.read\_input\_registers(1, 2).await?;   Ok(response) }   async fn send\_to\_server(  sensor\_id: &str,  location: &str,  process\_stage: &str,  temperature: f32,  humidity: f32,  timestamp: chrono::DateTime<Local>, ) -> Result<(), Box<dyn Error>> {  let mut stream = TcpStream::connect("127.0.0.1:7878").await?;   let payload = json!({  "timestamp": timestamp.to\_rfc3339\_opts(SecondsFormat::Secs, true),  "sensor\_id": sensor\_id,  "location": location,  "process\_stage": process\_stage,  "temperature\_celsius": temperature,  "humidity\_percent": humidity  });   let json\_str = payload.to\_string();  println!("Sending JSON: {}", json\_str);   // Send JSON payload  stream.write\_all(json\_str.as\_bytes()).await?;  stream.write\_all(b"\n").await?;    // Read response  let mut buf = [0; 1024];  let n = stream.read(&mut buf).await?;  println!("Server response: {}", std::str::from\_utf8(&buf[..n])?);   Ok(()) }  #[tokio::main] async fn main() -> Result<(), Box<dyn Error>> {  let sensor\_id = "SHT20-PascaPanen-001";  let location = "Gudang Fermentasi 1";  let process\_stage = "Fermentasi";   loop {  let timestamp = Local::now();   match sht20(1).await {  Ok(response) if response.len() == 2 => {  let temp = response[0] as f32 / 10.0;  let rh = response[1] as f32 / 10.0;   println!(  "[{}] {} - {}: Temp = {:.1}°C, RH = {:.1}%",  timestamp.format("%Y-%m-%d %H:%M:%S"),  location,  process\_stage,  temp,  rh  );   if let Err(e) = send\_to\_server(  sensor\_id,  location,  process\_stage,  temp,  rh,  timestamp,  )  .await  {  eprintln!("Failed to send data: {}", e);  }  }  Ok(invalid) => eprintln!("Invalid sensor response: {:?}", invalid),  Err(e) => eprintln!("Sensor read error: {}", e),  }   sleep(Duration::from\_secs(10)).await;  } } |

Cargo.toml

[package]  
name = "sht20"  
version = "0.1.0"  
edition = "2021"  
  
[dependencies]  
chrono = "0.4"  
serde\_json = "1.0"  
tokio = { version = "1.0", features = ["full"] }  
tokio-modbus = "0.9"  
tokio-serial = "5.4"

* 1. **Kode Rust TCP Server**

Tahap berikutnya adalah menerima data dari client menggunakan program Rust TCP Server. Server ini terus-menerus memonitor koneksi pada port USB. Ketika data JSON diterima, program memprosesnya dengan pustaka serde\_json untuk mengurai informasi yang dikirim oleh client. Setelah parsing, data disiapkan dan dikirimkan ke InfluxDB, sebuah database time-series yang dirancang untuk menyimpan data berurutan berdasarkan waktu. Data disimpan dengan struktur measurement untuk mencatat nama data, tags untuk metadata seperti ID sensor, dan fields untuk nilai-nilai aktual seperti suhu dan kelembaban.

Kode Rust TCP Server

**Main.rs**

use chrono::{DateTime, Utc};

use dotenvy::dotenv;

use influxdb2::{Client, models::DataPoint};

use serde::Deserialize;

use std::{env, time::Duration};

use tokio::{net::TcpListener, io::{AsyncReadExt, AsyncWriteExt}};

use futures::stream;

use modbus::{Client as ModbusClient, rtu};

#[derive(Debug, Deserialize)]

struct SensorData {

timestamp: String,

sensor\_id: String,

location: String,

process\_stage: String,

temperature\_celsius: f64,

humidity\_percent: f64,

}

#[tokio::main]

async fn main() -> Result<(), Box<dyn std::error::Error>> {

dotenv().ok();

let influx\_url = env::var("INFLUXDB\_URL")?;

let influx\_org = env::var("INFLUXDB\_ORG")?;

let influx\_token = env::var("INFLUXDB\_TOKEN")?;

let influx\_bucket = env::var("INFLUXDB\_BUCKET")?;

let client = Client::new(&influx\_url, &influx\_org, &influx\_token);

client.health().await?;

println!("✅ InfluxDB OK. Starting...");

let client\_modbus = client.clone();

let influx\_bucket\_modbus = influx\_bucket.clone();

tokio::spawn(async move {

loop {

match read\_sensor\_data("/dev/ttyUSB0", 1) {

Ok((temperature, humidity)) => {

let timestamp = Utc::now().timestamp\_nanos\_opt().unwrap\_or(0);

let point = DataPoint::builder("environment\_monitoring")

.tag("sensor\_id", "SHT40-01")

.tag("location", "Gudang A")

.tag("process\_stage", "fermentasi")

.field("temperature\_celsius", temperature)

.field("humidity\_percent", humidity)

.timestamp(timestamp)

.build()

.unwrap();

match client\_modbus.write(&influx\_bucket\_modbus, stream::iter(vec![point])).await {

Ok(\_) => println!("📡 Modbus data written to InfluxDB"),

Err(e) => eprintln!("❌ Modbus write error: {}", e),

}

}

Err(e) => eprintln!("❌ Modbus read error: {}", e),

}

tokio::time::sleep(Duration::from\_secs(10)).await;

}

});

// TCP Server

let client\_tcp = client.clone();

let influx\_bucket\_tcp = influx\_bucket.clone();

let listener = TcpListener::bind("127.0.0.1:7878").await?;

println!("🔌 TCP Server listening on 127.0.0.1:7878");

loop {

let (mut socket, \_) = listener.accept().await?;

let client = client\_tcp.clone();

let bucket = influx\_bucket\_tcp.clone();

tokio::spawn(async move {

let mut buf = [0; 1024];

loop {

let n = match socket.read(&mut buf).await {

Ok(0) => break,

Ok(n) => n,

Err(e) => {

eprintln!("Socket error: {}", e);

break;

}

};

let data = match std::str::from\_utf8(&buf[..n]) {

Ok(d) => d,

Err(\_) => {

let \_ = socket.write\_all(b"ERROR: Invalid UTF-8").await;

continue;

}

};

match serde\_json::from\_str::<SensorData>(data) {

Ok(sensor) => {

let timestamp = match DateTime::parse\_from\_rfc3339(&sensor.timestamp) {

Ok(dt) => dt.with\_timezone(&Utc).timestamp\_nanos\_opt().unwrap\_or(0),

Err(\_) => {

let \_ = socket.write\_all(b"ERROR: Timestamp format").await;

continue;

}

};

let point = DataPoint::builder("environment\_monitoring")

.tag("sensor\_id", &sensor.sensor\_id)

.tag("location", &sensor.location)

.tag("process\_stage", &sensor.process\_stage)

.field("temperature\_celsius", sensor.temperature\_celsius)

.field("humidity\_percent", sensor.humidity\_percent)

.timestamp(timestamp)

.build()

.unwrap();

match client.write(&bucket, stream::iter(vec![point])).await {

Ok(\_) => {

let \_ = socket.write\_all(b"OK").await;

println!("📥 TCP data written to InfluxDB");

}

Err(e) => {

let \_ = socket.write\_all(b"ERROR: InfluxDB write").await;

eprintln!("❌ TCP write error: {}", e);

}

}

}

Err(e) => {

let \_ = socket.write\_all(b"ERROR: JSON format").await;

eprintln!("❌ TCP parse error: {}", e);

}

}

}

});

}

}

// ✅ Modbus RTU read

fn read\_sensor\_data(port: &str, slave: u8) -> Result<(f64, f64), Box<dyn std::error::Error + Send + Sync>> {

let mut ctx = rtu::connect\_slave(port, slave)?;

let regs = ctx.read\_input\_registers(0x0000, 2)?; // Ganti offset jika beda

let temp = regs[0] as f64 / 100.0;

let humid = regs[1] as f64 / 100.0;

Ok((temp, humid))

}

**Cargo.toml**

[package]

name = "tcp\_server"

version = "0.1.0"

edition = "2021"

[dependencies]

tokio = { version = "1", features = ["full"] }

serde = { version = "1.0", features = ["derive"] }

serde\_json = "1.0" # Tambahkan ini

influxdb2 = "0.5"

chrono = { version = "0.4", features = ["serde"] }

futures = "0.3"

dotenvy = "0.15"

modbus = "1.1.1"

**.env**

INFLUXDB\_URL=http://localhost:8086

INFLUXDB\_ORG=BLITAR

INFLUXDB\_TOKEN=UuSq6wTkE1tknOR5zYAVWxij-yzvUlrnokSq9LFlxG5HTysteFX6bHxg\_rhTZudZGwwNZ0Y9BJlPdmDhY9cOnw==

INFLUXDB\_BUCKET=Sensor\_Temperature

* 1. **Konfigurasi InfluxDB dan Integrasi**

Tahap berikutnya adalah menerima data dari client menggunakan program Rust TCP Server. Server ini terus-menerus memonitor koneksi pada port USB. Ketika data JSON diterima, program memprosesnya dengan pustaka serde\_json untuk mengurai informasi yang dikirim oleh client. Setelah parsing, data disiapkan dan dikirimkan ke InfluxDB, sebuah database time-series yang dirancang untuk menyimpan data berurutan berdasarkan waktu. Data disimpan dengan struktur measurement untuk mencatat nama data, tags untuk metadata seperti ID sensor, dan fields untuk nilai-nilai aktual seperti suhu dan kelembaban.

Hasil Query di InfluxDB CLI

|  |
| --- |
| #include "influxreader.h" #include #include  InfluxReader::InfluxReader(QObject \*parent) : QObject(parent) { manager = new QNetworkAccessManager(this); connect(manager, &QNetworkAccessManager::finished, this, &InfluxReader::onReply); }  void InfluxReader::queryInflux() { QString org = "BLITAR"; QString token = "YOUR\_TOKEN\_HERE"; // ganti token Anda QString url = "[http://localhost:8086/api/v2/query?org="](http://localhost:8086/api/v2/query?org=%22) + org;  QNetworkRequest request(QUrl(url)); request.setRawHeader("Authorization", ("Token " + token).toUtf8()); request.setHeader(QNetworkRequest::ContentTypeHeader, "application/vnd.flux");  QString fluxQuery = "from(bucket: \"pertanian\") |> range(start: -1m)";  manager->post(request, fluxQuery.toUtf8());  }  void InfluxReader::onReply(QNetworkReply \*reply) { if (reply->error() != QNetworkReply::NoError) { qDebug() << "Error:" << reply->errorString(); reply->deleteLater(); return; }  QByteArray responseData = reply->readAll(); qDebug() << "Response from InfluxDB:" << responseData; emit dataReceived(responseData);  reply->deleteLater();  }  #ifndef INFLUXREADER\_H #define INFLUXREADER\_H  #include #include #include  class InfluxReader : public QObject { Q\_OBJECT  public: explicit InfluxReader(QObject \*parent = nullptr); void queryInflux();  signals: void dataReceived(const QByteArray &data);  private slots: void onReply(QNetworkReply \*reply);  private: QNetworkAccessManager \*manager; };  #endif // INFLUXREADER\_H #ifndef INFLUXREADER\_H #define INFLUXREADER\_H  #include #include #include  class InfluxReader : public QObject { Q\_OBJECT  public: explicit InfluxReader(QObject \*parent = nullptr); void queryInflux();  signals: void dataReceived(const QByteArray &data);  private slots: void onReply(QNetworkReply \*reply);  private: QNetworkAccessManager \*manager; };  #endif // INFLUXREADER\_H |

* 1. **Dashboard Grafana**

Tahap berikutnya adalah menerima data dari client menggunakan program Rust TCP Server. Server ini terus-menerus memonitor koneksi pada port USB. Ketika data JSON diterima, program memprosesnya dengan pustaka serde\_json untuk mengurai informasi yang dikirim oleh client. Setelah parsing, data disiapkan dan dikirimkan ke InfluxDB, sebuah database time-series yang dirancang untuk menyimpan data berurutan berdasarkan waktu. Data disimpan dengan struktur measurement untuk mencatat nama data, tags untuk metadata seperti ID sensor, dan fields untuk nilai-nilai aktual seperti suhu dan kelembaban.

Dashboard Grafana

|  |
| --- |
|  |

* 1. **Dashboard QT**

Connecting ke InfluxDB

|  |
| --- |
|  |

Struktur Program

|  |
| --- |
|  |

**BAB IV**

**HASIL PENELITIAN**

* + 1. Hasil Grafana

|  |
| --- |
|  |

* + 1. Hasil Real-time Dashboard InfluxDB

|  |
| --- |
| A screenshot of a computer  AI-generated content may be incorrect. |

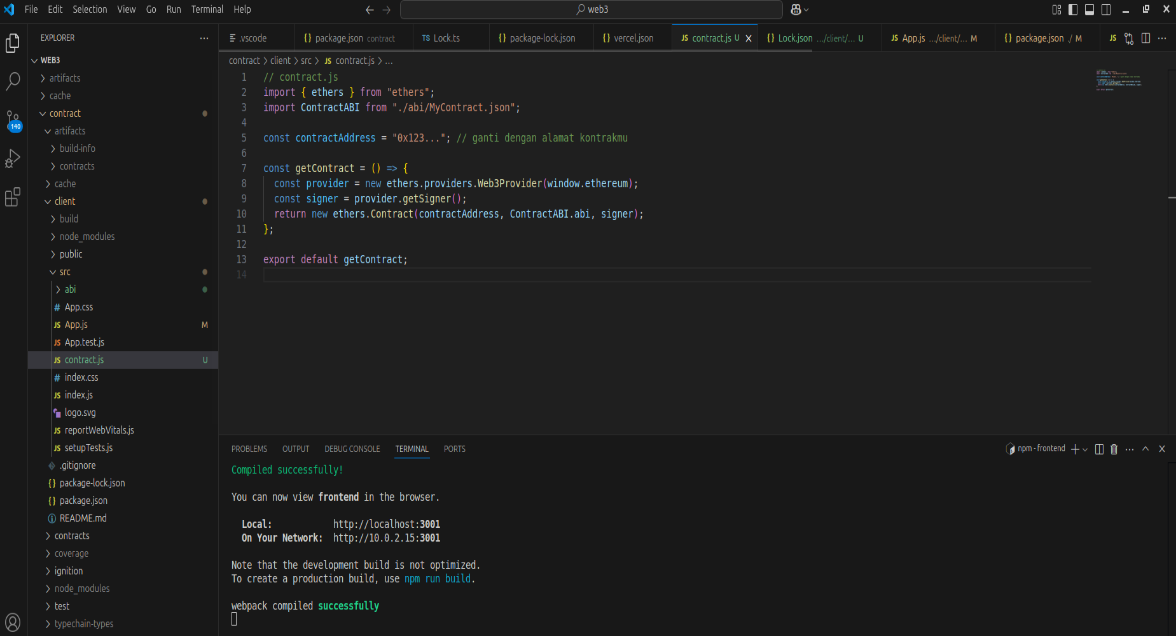
* + 1. Hasil Real-time Dashboard Qt

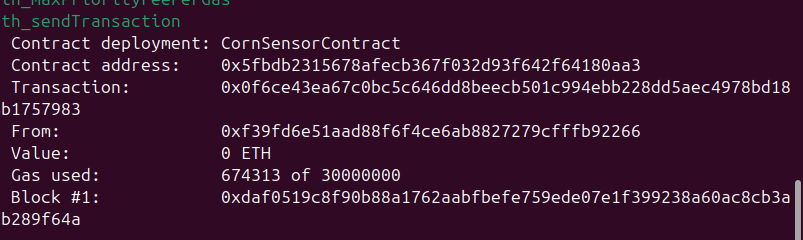
|  |
| --- |
| A screen shot of a graph  AI-generated content may be incorrect. |

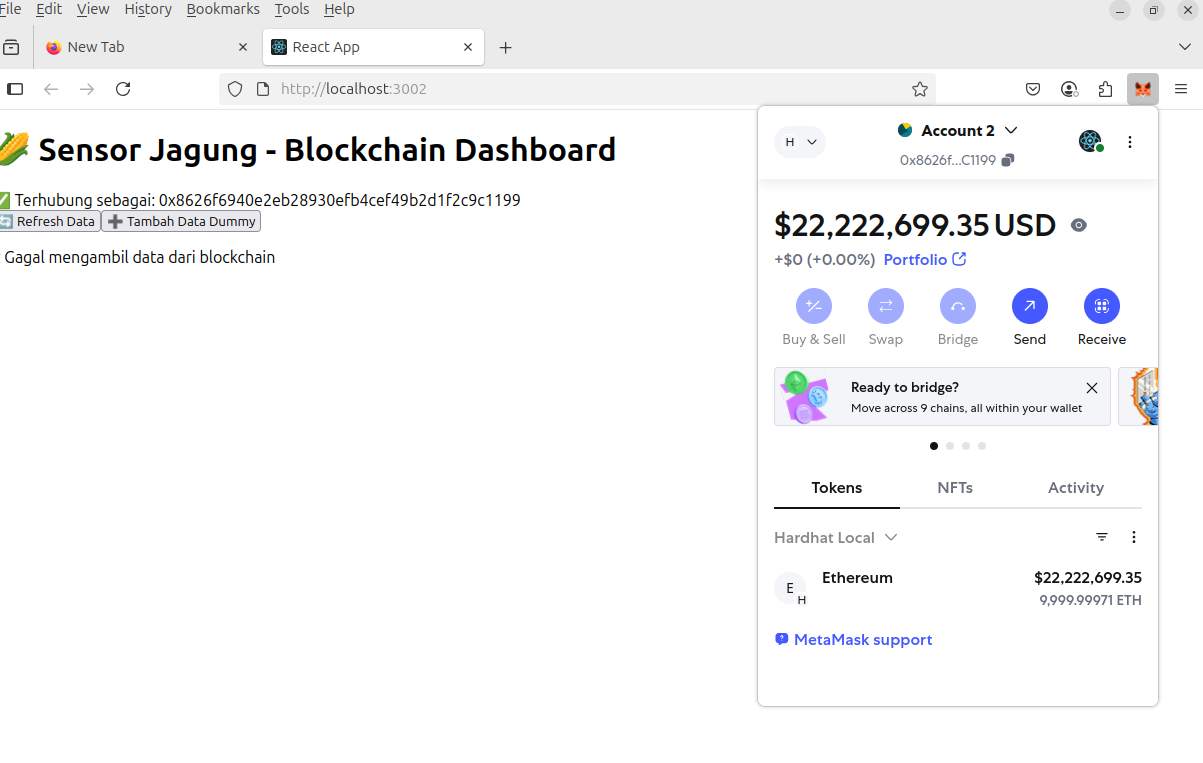
* + 1. Hasil Metamask

|  |
| --- |
| A computer screen shot of a computer  AI-generated content may be incorrect.  A white background with black dots  AI-generated content may be incorrect. |

* 1. **Hasil Real-time Dashboard Web3**
  2. **Hasil Blockchain**

****

****

****

**BAB V**

**PENUTUP**

**Kesimpulan**

Proyek CORNTECH berhasil mengembangkan sebuah sistem monitoring suhu dan kelembaban yang inovatif untuk gudang penyimpanan jagung, dengan memadukan teknologi Internet of Things (IoT), blockchain Ethereum, InfluxDB, dan Grafana. Sistem ini dirancang untuk menjaga kualitas komoditas pertanian selama masa penyimpanan dengan menyediakan pemantauan lingkungan yang akurat dan real-time. Data diperoleh dari sensor SHT20 yang memantau suhu dan kelembaban secara berkelanjutan, kemudian disimpan secara efisien menggunakan InfluxDB sebagai basis data time-series. Visualisasi dilakukan melalui dua jalur: antarmuka lokal berbasis Qt dan dashboard daring melalui Grafana, sehingga pengguna dapat mengakses data secara fleksibel baik secara offline maupun online. Salah satu keunggulan utama CORNTECH adalah penerapan teknologi blockchain untuk menjamin integritas dan keaslian data. Ringkasan data lingkungan yang mencakup nilai rata-rata, maksimum, dan minimum dikonversi menjadi nilai hash dan dicatat ke dalam smart contract di jaringan Ethereum.

Proses ini memastikan bahwa data tidak dapat diubah dan dapat diverifikasi secara publik, meningkatkan transparansi serta kepercayaan para pemangku kepentingan. Untuk mendukung proses ini, sistem juga dilengkapi dengan Decentralized Application (DApp) berbasis Web3.js dan MetaMask yang memungkinkan distributor, pengelola gudang, atau konsumen akhir melakukan verifikasi hash data secara langsung dari blockchain. Dengan mengintegrasikan sistem digital monitoring berbasis IoT dan teknologi Web3, CORNTECH tidak hanya meningkatkan efisiensi dan keandalan pengelolaan gudang, tetapi juga memberikan nilai tambah melalui jaminan keaslian dan keterbukaan data. Sistem ini berkontribusi secara signifikan terhadap upaya pengurangan kerugian pangan pasca-panen, peningkatan kualitas penyimpanan komoditas pertanian, dan membangun kepercayaan konsumen terhadap rantai pasok pangan nasional yang lebih cerdas dan transparan.

**Saran**

Proyek CORNTECH telah berhasil mewujudkan sistem monitoring suhu dan kelembaban berbasis teknologi IoT dan blockchain untuk gudang penyimpanan beras. Dengan menggabungkan sensor industri, database time-series InfluxDB, visualisasi real-time melalui Grafana dan Qt, serta verifikasi keaslian data menggunakan smart contract di jaringan Ethereum, sistem ini menjadi fondasi yang kuat dalam memastikan kualitas penyimpanan komoditas pertanian. Meski telah berjalan sesuai tema dan memberikan hasil yang menjanjikan, masih terdapat sejumlah peluang pengembangan dan saran untuk peningkatan ke depan. Pertama, untuk meningkatkan keakuratan dan cakupan pemantauan lingkungan, sistem ini dapat diperluas dengan menambahkan sensor-sensor tambahan seperti detektor tekanan, gas, dan kelembaban tanah. Hal ini membuka peluang adopsi sistem ke sektor pertanian lapangan terbuka, gudang bahan makanan lain, atau area penyimpanan berisiko tinggi. Kedua, meskipun blockchain sudah memberikan keamanan data yang tinggi, implementasi sistem enkripsi tambahan serta opsi penggunaan private atau hybrid blockchain akan lebih menjamin kerahasiaan dan kendali atas data sensitif. Ini penting, terutama jika sistem digunakan oleh berbagai entitas dalam skala besar.

Ketiga, perlu adanya peningkatan pada aspek skalabilitas dan manajemen jaringan, mengingat potensi penerapan CORNTECH di jaringan gudang besar dengan banyak sensor dan lokasi. Dukungan untuk integrasi sistem pelacakan logistik juga menjadi nilai tambah, terutama jika sistem dikembangkan untuk mendukung distribusi barang sensitif seperti obat-obatan atau makanan beku. Terakhir, diperlukan pengujian lanjutan secara lebih luas di berbagai lokasi penyimpanan guna menguji keandalan sistem dalam kondisi nyata, serta mengevaluasi aspek ketahanan dan efisiensinya dalam jangka panjang. Dengan pengembangan berkelanjutan, CORNTECH berpotensi menjadi solusi teknologi yang mendukung keberlanjutan, efisiensi, dan transparansi dalam rantai pasok pangan maupun sektor logistik lainnya.

**DAFTAR PUSTAKA**

Altmeyer, S., & Konietzko, J. (2018). Qt-Based SCADA System for Industrial Applications. Journal of Industrial Control and Automation, 9(2), 56–61.

Babu, G. V., & Rao, M. M. (2022). Real-Time Smart Monitoring of Temperature and Humidity Using InfluxDB and Grafana. International Journal of Engineering and Advanced Technology, 11(3), 35–40.

Barksdale, A., & Stone, T. (2019). *Scalable Time-Series Data Infrastructure Using InfluxDB for IoT Sensor Monitoring*. *Journal of Sensor Networks*, 5(4), 202–210.

Blanchette, J., & Summerfield, M. (2006). C++ GUI Programming with Qt 4 (2nd ed.). Prentice Hall.

Buterin, V. (2020). The Meaning of Decentralization. Ethereum Foundation. https://ethereum.org/en/developers/docs/intro-to-web3/

Casino, F., Dasaklis, T. K., & Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. Telecommunications Policy, 43(10), 101848. https://doi.org/10.1016/j.telpol.2019.101848

CNN Indonesia. (2024, February 5). *Bulog Akui Gudang Beras Sering Kelembaban Tinggi, Kualitas Menurun*. https://www.cnnindonesia.com/ekonomi/bulog-akui-kualitas-beras-menurun-karena-gudang-lembap

Dahalin, M., et al. (2021). IoT-Based Real-Time Environmental Monitoring System Using MQTT, InfluxDB, and Grafana. Indonesian Journal of Electrical Engineering and Computer Science, 22(1), 151–158. https://doi.org/10.11591/ijeecs.v22.i1.pp151-158

FAO. (2019). *Postharvest Management of Rice and Grains*. Food and Agriculture Organization of the United Nations.

Gupta, R., & Singh, H. (2020). Real-time IoT Dashboard Design Using Grafana and InfluxDB. International Journal of Computer Applications, 176(22), 1–6. https://doi.org/10.5120/ijca2020919934

Kementerian Pertanian RI. (2023). *Transformasi Pertanian Menuju Agriculture 4.0*. Jakarta: Pusat Data dan Sistem Informasi Pertanian.

Kumar, R., Singh, A., & Sharma, M. (2020). Implementation of Real-Time Industrial Data Monitoring System Using Modbus RTU Protocol. *Journal of Industrial Automation*, 12(3), 45–52.