

**PROJECT MATA KULIAH**  
**INTERKONEKSI SITEM INSTRUMENTASI**

Dosen : Ahmad Radhy, S.Si., M.Si.

**“Pemantauan Proses Fermentasi Tape Berbasis IoT dan Blockchain untuk  
Menjaga Kualitas Produk melalui Monitoring Suhu dan Kelembapan”**



Disusun Oleh :

Praganta Walhidaya	2042231040
Maulana Ahmad Dzulqornain	2042231070
Anif Maulana Ramadhon	2042231074

Kelas 4B

**PRODI D4 TEKNOLOGI REKAYASA INSTRUMENTASI**  
**DEPARTMENT TEKNIK INSTRUMENTASI**  
**FAKULTAS VOKASI**  
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**2025**

# **BAB I**

## **PENDAHULUAN**

### **Latar Belakang**

Tape merupakan produk fermentasi tradisional Indonesia yang dihasilkan dari bahan dasar singkong atau ketan yang difermentasi menggunakan ragi (mikroorganisme seperti *Saccharomyces cerevisiae* dan *Amylomyces rouxii*). Proses fermentasi tape sangat dipengaruhi oleh faktor lingkungan seperti suhu dan kelembapan. Kondisi yang tidak stabil selama fermentasi dapat memengaruhi rasa, tekstur, kadar alkohol, serta keamanan produk. Suhu yang terlalu tinggi dapat mempercepat proses fermentasi secara berlebihan, sementara kelembapan yang terlalu rendah atau terlalu tinggi dapat menyebabkan tape menjadi terlalu kering atau justru busuk. Oleh karena itu, pengawasan terhadap parameter lingkungan sangat penting untuk menjaga kualitas dan konsistensi tape.

Di tengah meningkatnya minat masyarakat terhadap produk lokal dan organik, tape memiliki potensi pasar yang luas, baik secara tradisional maupun digital. Untuk meningkatkan efisiensi produksi dan memperluas distribusi, teknologi digital dapat diintegrasikan dalam proses pembuatan dan penjualan tape. Sistem monitoring berbasis Internet of Things (IoT) diterapkan untuk memantau suhu dan kelembapan secara real-time selama proses fermentasi. Di sisi lain, sistem penjualan tape akan berbasis website yang terhubung dengan teknologi Web3 dan smart contract. Teknologi ini menyediakan transaksi yang transparan, aman, dan tidak bergantung pada perantara. Smart contract akan memastikan bahwa setiap transaksi tereksekusi otomatis sesuai kesepakatan digital, menciptakan ekosistem perdagangan tape yang lebih modern dan terpercaya.

Dengan kombinasi sistem monitoring fermentasi dan transaksi digital berbasis blockchain, proyek ini menghadirkan inovasi pada produk pangan lokal Indonesia. Pendekatan ini tidak hanya meningkatkan kualitas dan efisiensi produksi tape, tetapi juga memperkenalkan pendekatan pemasaran dan distribusi yang sesuai dengan perkembangan teknologi masa kini.

### **1.1 Rumusan Masalah**

1. Bagaimana merancang sistem monitoring suhu dan kelembapan secara real-time pada proses fermentasi tape ?
2. Bagaimana cara menyimpan dan menampilkan data suhu dan kelembapan sebagai acuan untuk menjaga kualitas tape ?
3. Bagaimana merancang sistem penjualan tape melalui website yang terintegrasi dengan Web3 ?

### **1.2 Tujuan Masalah**

1. Merancang dan membangun sistem monitoring suhu dan kelembapan secara real-time yang dapat digunakan untuk memantau kondisi lingkungan selama proses fermentasi tape.

2. Mengembangkan sistem penyimpanan dan visualisasi data suhu dan kelembapan agar dapat digunakan sebagai acuan untuk menjaga dan meningkatkan kualitas tape yang dihasilkan.
3. Merancang platform penjualan tape berbasis website yang terintegrasi dengan teknologi Web3, termasuk sistem pembayaran menggunakan smart contract untuk mendukung transaksi yang transparan dan aman.

### **1.3 Manfaat**

1. Bagi Masyarakat: Proyek ini memberikan dorongan terhadap pelestarian dan peningkatan nilai ekonomi dari produk fermentasi lokal seperti tape. Dengan adanya sistem monitoring dan distribusi digital, masyarakat, khususnya pelaku UMKM, dapat meningkatkan kualitas produksi tape secara konsisten serta memperluas jangkauan pemasaran hingga ke pasar digital berbasis Web3. Hal ini turut mendukung pemberdayaan ekonomi lokal dan transformasi digital di sektor pangan tradisional.
2. Bagi Konsumen: Mendapatkan jaminan produk tape yang berkualitas dan terjaga proses produksinya. Melalui transparansi data fermentasi dan sistem transaksi berbasis smart contract, konsumen dapat lebih percaya terhadap keamanan, keaslian, serta keadilan dalam proses jual beli. Selain itu, konsumen juga diperkenalkan pada pengalaman berbelanja berbasis teknologi blockchain yang modern dan aman.
3. Bagi Mahasiswa: Proyek ini menjadi media pembelajaran langsung bagi mahasiswa dalam menggabungkan teknologi IoT, blockchain, dan Web3 dalam satu solusi nyata. Mahasiswa tidak hanya memahami konsep teoritis, tetapi juga menerapkannya secara praktis pada bidang pangan dan kewirausahaan digital. Hal ini mendorong pengembangan kemampuan lintas disiplin yang sangat relevan dengan tantangan industri masa kini dan masa depan.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 SHT 20

Sensor SHT20 merupakan sensor suhu dan kelembaban digital yang banyak digunakan dalam aplikasi industri karena memiliki tingkat akurasi tinggi dan stabilitas yang baik. Sensor ini menggunakan protokol komunikasi Modbus RTU yang berjalan melalui jalur komunikasi RS485, sehingga memungkinkan untuk diterapkan pada jarak transmisi data yang cukup jauh dan dapat dihubungkan secara multidrop. SHT20 mampu mengukur suhu dalam rentang  $-40^{\circ}\text{C}$  hingga  $125^{\circ}\text{C}$  dengan akurasi  $\pm 0.3^{\circ}\text{C}$ , serta kelembaban relatif (RH) dalam rentang 0% hingga 100% dengan akurasi  $\pm 3\%\text{RH}$ . Sensor ini dipilih karena tahan terhadap lingkungan industri dan mampu memberikan hasil pembacaan yang presisi untuk kebutuhan monitoring ruang produksi maupun penyimpanan (Priyanto et al., 2024).



*Gambar 2.1 Modbus SHT20 Sensor*

#### 2.2 Converter RS485

Modbus Client merupakan program yang dikembangkan menggunakan bahasa pemrograman Rust dan berfungsi sebagai master yang melakukan komunikasi dengan sensor SHT20 melalui protokol Modbus RTU (Sharma et al., 2023). Program ini bertugas untuk membaca nilai suhu dan kelembaban dari sensor secara periodik, memproses data tersebut, kemudian membentuk payload dalam format JSON yang berisi informasi hasil pembacaan, waktu, lokasi sensor, ID sensor, dan tahap proses produksi. Payload JSON ini selanjutnya dikirimkan ke TCP Server melalui koneksi TCP socket. Pemanfaatan Rust dipilih karena memiliki performa tinggi, aman terhadap memory leak, dan cocok untuk aplikasi sistem monitoring berbasis jaringan.



***Gambar 2.2 Converter RS485***

### **2.3 Ubuntu**

Ubuntu adalah sistem operasi open-source berbasis kernel Linux yang dikembangkan oleh Canonical Ltd. Sistem ini dirancang untuk memberikan kemudahan penggunaan, keamanan, serta stabilitas, sehingga menjadi pilihan populer di berbagai bidang mulai dari pendidikan, pengembangan perangkat lunak, hingga industri. Ubuntu memiliki beberapa versi seperti Ubuntu Desktop, Ubuntu Server, dan Ubuntu Core yang masing-masing disesuaikan untuk keperluan umum, server, dan sistem tertanam atau IoT. Ubuntu banyak digunakan dalam pengembangan sistem monitoring dan otomasi karena kompatibel dengan berbagai bahasa pemrograman seperti Python, C++, dan Rust. Selain itu, Ubuntu mendukung instalasi dan pengelolaan berbagai perangkat lunak open-source seperti InfluxDB, Grafana, dan Mosquitto MQTT yang sering digunakan untuk mencatat, menyimpan, dan menampilkan data sensor secara real-time. Dengan arsitektur yang ringan dan stabil, Ubuntu cocok untuk sistem pengawasan suhu dan kelembapan di lingkungan industri maupun laboratorium. Keunggulan lain dari Ubuntu adalah tersedianya dukungan jangka panjang (LTS), sistem keamanan berlapis seperti AppArmor, serta komunitas pengembang aktif yang mempercepat proses troubleshooting dan pengembangan. Karena itu, Ubuntu menjadi fondasi sistem operasi yang sangat handal untuk proyek-proyek monitoring berbasis TCP/IP, termasuk untuk pemantauan suhu dan kelembapan dalam penyimpanan bahan pangan seperti tape.



***Gambar 2.3 Ubuntu***

### **2.4 Rust**

TCP Server juga merupakan program yang dibuat menggunakan Rust dan berperan sebagai penerima data dari Modbus Client. Program ini akan

menerima paketdata dalam format JSON melalui koneksi TCP, melakukan proses parsing ataupun penguraian isi data, serta memvalidasi informasi yang diterima. Setelah data berhasil diproses, server akan meneruskannya ke InfluxDB melalui API HTTP dengan format penulisan yang sesuai dengan struktur database time-series. TCP Server ini bertanggung jawab memastikan data yang masuk sesuai standar dan langsung diteruskan ke sistem penyimpanan tanpa kehilangan atau kerusakan data (Sopasakis et al., 2020).



*Gambar 2.4 Rust Program*

## 2.5 InfluxDb

InfluxDB adalah database khusus time-series yang dirancang untuk menyimpan data berdasarkan waktu pengambilan, sehingga sangat sesuai untuk menyimpan data sensor yang dikirimkan secara periodik (Nasar & Kausar, 2019). InfluxDB menyimpan informasi berupa timestamp, nilai suhu, nilai kelembaban, ID sensor, lokasi, dan tahap proses produksi dalam format yang terstruktur dan dapat diakses dengan cepat. Database ini memungkinkan penyimpanan data dalam jumlah besar tanpa mengurangi kecepatan query, sehingga cocok digunakan untuk sistem monitoring lingkungan industri yang memerlukan data historis sebagai bahan analisis maupun pelaporan.



*Gambar 2.5 InfluxDB*

## 2.6 Grafana

Grafana merupakan platform visualisasi data open-source yang dapat diintegrasikan dengan berbagai jenis database, salah satunya adalah InfluxDB. Pada sistem ini, Grafana berfungsi menampilkan data suhu dan kelembaban yang disimpan di InfluxDB dalam bentuk visual yang mudah dipahami, seperti grafik garis (time-series chart), gauge meter, tabel, hingga notifikasi peringatan

jika nilai suhu atau kelembaban melebihi ambang batas (Aggoune & Benratem, 2023). Dashboard yang dibuat pada Grafana dapat diakses secara real-time melalui browser, sehingga memudahkan operator atau manajer produksi dalam melakukan monitoring kondisi lingkungan di area industri secara cepat dan akurat.



***Gambar 2.6 Grafana***

## **2.7 Qt**

Qt adalah framework pengembangan aplikasi lintas platform yang digunakan untuk membuat antarmuka pengguna grafis (GUI) yang responsif dan interaktif. Dalam sistem monitoring ini, Qt berperan sebagai platform untuk mengembangkan aplikasi desktop atau mobile yang mampu menampilkan data suhu dan kelembaban secara real-time dengan tampilan yang dapat dikustomisasi sesuai kebutuhan pengguna (Nabijonovich & Najmiddin, 2024). Aplikasi berbasis Qt memungkinkan integrasi langsung dengan database InfluxDB untuk mengambil data time-series dan menyajikannya dalam bentuk grafik, indikator, atau tabel yang mudah dipahami. Keunggulan Qt terletak pada fleksibilitas desain UI dan performa aplikasi yang stabil, sehingga memudahkan petani atau operator untuk mengakses informasi secara offline maupun online dengan pengalaman pengguna yang nyaman.



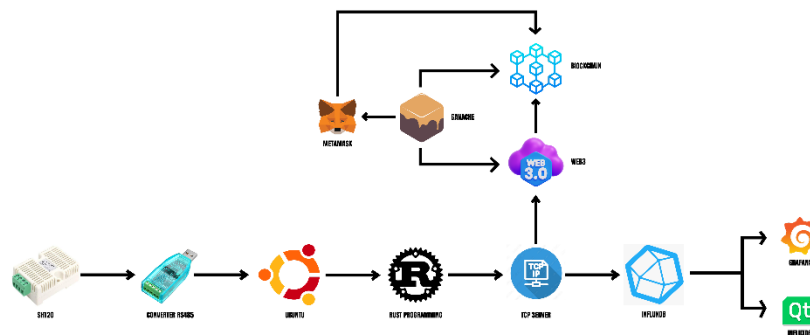
***Gambar 2.7 Grafana***

## BAB III TAHAP PELAKSANAAN

### 3.1 Studi Literatur

Pada studi literatur dilakukan proses untuk mencari referensi-referensi yang berkaitan dengan fermentasi tape berbasis Web3. Sumber referensi sangat beragam mulai dari artikel, jurnal, paper, buku, atau hasil penelitian lainnya. Materi yang telah didapatkan diidentifikasi dan dilakukan perumusan masalah, dari perumusan masalah dapat ditarik tujuan dari penelitian ini.

### 3.2 Perancangan Sistem



**Gambar 3.1** Alur Sistem Kerja

### 3.3 Pembuatan Program

Pembuatan program dalam proyek ini terdiri dari tiga bagian utama, yaitu:

1. Program Client Sensor (Embedded Rust): membaca data suhu dan kelembaban melalui Modbus RTU.
2. Program Server (Rust): menerima data via TCP, menyimpan ke InfluxDB, dan mencatat ke Blockchain.
3. Antarmuka Visualisasi: terdiri dari GUI desktop berbasis Python dan dashboard Web3 berbasis HTML–JS.

Setiap bagian dirancang untuk berjalan secara sinkron dan saling terhubung melalui protokol terbuka, seperti TCP, HTTP, dan Web3 RPC.

### 3.4 Pembuatan Program Client Sensor

Program client dikembangkan menggunakan bahasa Rust dengan pustaka tokio-modbus untuk komunikasi Modbus RTU melalui port serial (/dev/ttyUSB0). Sensor SHT20 dikonfigurasi untuk mengirimkan dua register yang masing-masing merepresentasikan suhu dan kelembaban. Data dibaca setiap 5 detik, dikonversi dari satuan raw register menjadi suhu dalam °C dan kelembaban dalam %RH, lalu dikemas dalam format JSON.

Data yang sudah terstruktur dikirim ke server melalui TCP socket menggunakan tokio::net::TcpStream. Jika koneksi berhasil, JSON dikirim disertai newline (\n) sebagai delimiter. Sistem mampu menangani connection failure dan data loss dengan retry otomatis setiap siklus pembacaan.



### 3.5 Pembuatan Program Server

Server dikembangkan dengan Rust asynchronous menggunakan tokio dan reqwest. Ia menerima koneksi masuk pada port 9000, membaca data sensor, dan melakukan dua proses:

1. Penyimpanan InfluxDB: Data dikonversi ke format Line Protocol, lalu dikirim melalui HTTP POST ke endpoint API InfluxDB v2.
2. Pencatatan ke Blockchain: Smart contract SensorStorage.sol di-deploy secara otomatis menggunakan ethers-rs. Metode storeData() dipanggil untuk menyimpan data sensor ke dalam blockchain Ethereum lokal (localhost:8545) melalui Web3 JSON-RPC. Program ini memanfaatkan struktur JSON deserialisasi otomatis (serde) dan parsing timestamp RFC3339 ke epoch UNIX untuk integrasi dengan time-series database.

### 3.6 Pembuatan GUI Desktop

GUI desktop dikembangkan menggunakan Python dengan Tkinter dan matplotlib. Program ini memanggil data dari InfluxDB menggunakan query FLUX melalui endpoint /api/v2/query. Data suhu dan kelembaban ditampilkan dalam format numerik dan grafik waktu nyata (real-time chart).

Fitur utama :

- a. Tampilan suhu & kelembaban secara langsung.
- b. Grafik historis 50 titik data terakhir.
- c. Input manual untuk melihat data historis dengan rentang waktu tertentu.

Proses update dilakukan dalam thread paralel untuk memastikan GUI tetap responsif selama pengambilan data dari InfluxDB.

### 3.7 Pembuatan Web3 Dashboard

Antarmuka Web3 dikembangkan menggunakan HTML, Chart.js, dan Ethers.js. Dashboard ini menampilkan:

- a. Tabel log data sensor dari blockchain.
- b. Grafik suhu dan kelembaban dari event blockchain.
- c. Simulasi pemesanan container dengan validasi suhu & kelembaban sesuai kriteria.

Dashboard terhubung langsung ke smart contract SensorStorage menggunakan alamat kontrak dan ABI yang telah di-fetch. Fitur tambahan seperti QR simulasi pembelian ditampilkan setelah data sesuai ditemukan di blockchain.

### 3.8 Dependensi dan Konfigurasi

Seluruh dependensi untuk proyek ini didefinisikan dalam file Cargo.toml, package.json, dan requirements.txt. Untuk Rust, digunakan pustaka berikut :

- tokio, tokio-serial, tokio-modbus (I/O dan komunikasi)
- serde, serde\_json (serialisasi JSON)
- reqwest (HTTP client untuk InfluxDB)
- ethers (integrasi Web3/Ethereum)

- chrono (manajemen waktu)

### 3.9 Pengujian System

Pengujian sistem bertujuan untuk memastikan bahwa setiap komponen dalam sistem monitoring suhu dan kelembaban dapat berfungsi dengan baik sesuai rancangan, mulai dari pembacaan sensor, komunikasi data, penyimpanan di database, visualisasi data, hingga pencatatan ke dalam blockchain. Pengujian ini juga dilakukan untuk mengevaluasi performa sistem dalam kondisi operasional nyata di lingkungan midstream migas.

Pengujian dilakukan secara berjenjang dan terpisah berdasarkan blok fungsi sistem, dengan metode sebagai berikut:

1. Uji TCP *Transmission*: menguji keberhasilan pengiriman data dari *client* ke server menggunakan protokol TCP/IP.
2. Uji Penyimpanan dan Visualisasi Data: mengecek apakah data berhasil disimpan ke InfluxDB dan divisualisasikan melalui Grafana dan PyQt GUI.
3. Uji Pencatatan ke *Blockchain*: menguji apakah data sensor berhasil tercatat ke dalam *smart contract* Ethereum dan dapat diakses melalui DApp berbasis Web3.

#### 3.3.1 Metodologi Pengujian

Pengujian dilakukan berdasarkan tahapan dan komponen berikut:

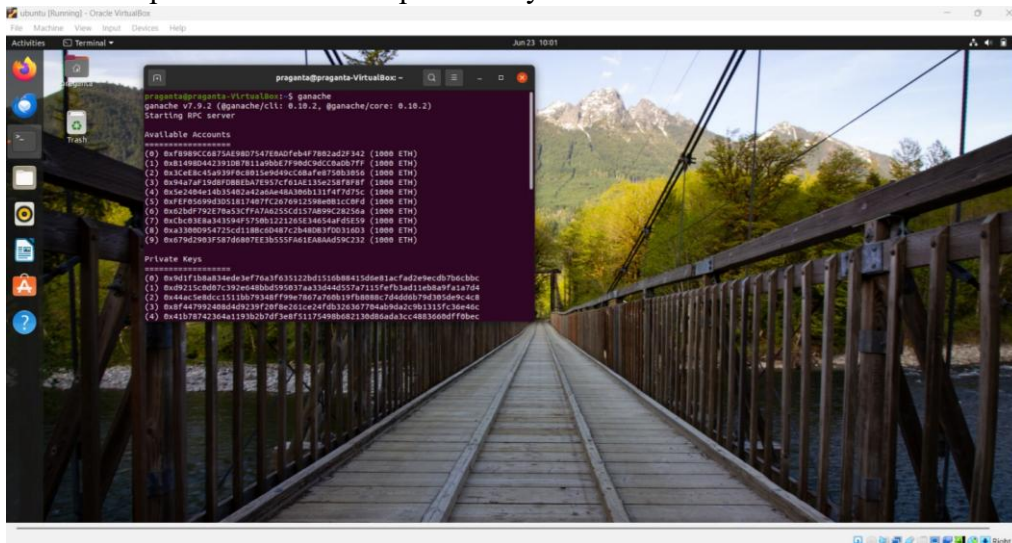
Komponen	Tujuan	Metode Pengujian
Client Sensor (Rust)	Memastikan sensor membaca suhu & kelembaban	Dibaca melalui Modbus RTU (RS485)
Komunikasi TCP	Memastikan data terkirim dari client ke server	TCP stream ditest setiap 5 detik
Server (Rust)	Parsing JSON, kirim ke InfluxDB & Smart Contract	Monitor via log terminal & Grafana
InfluxDB	Menyimpan data time-series	Dicek via dashboard & query FLUX
Ethereum Smart Contract	Mencatat log sensor	Dicek via hash transaksi dan DApp
PyQt GUI	Visualisasi real-time & historis	Dibandingkan dengan data aslinya
Web3 Dashboard	Tampilkan data & simulasi	Verifikasi tabel & chart sensor

## BAB IV

### PEMBAHASAN DAN HASIL

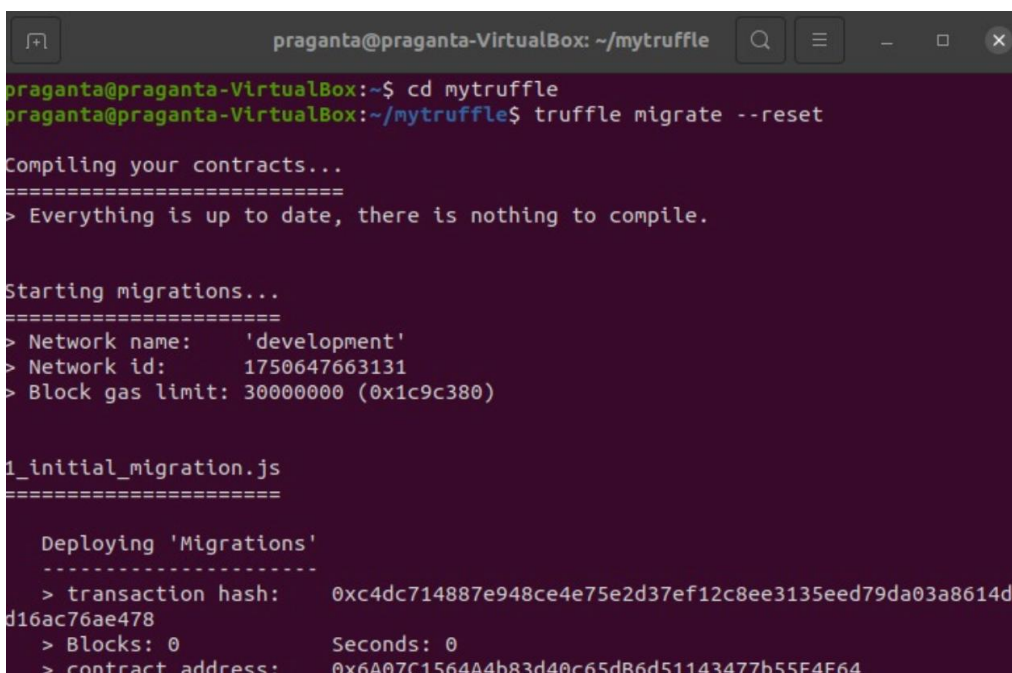
#### 4.1 Langkah Kerja

1. Buka Terminal lalu jalankan ganache dengan prompt `> ganache >` maka akan menampilkan account dan private keys.



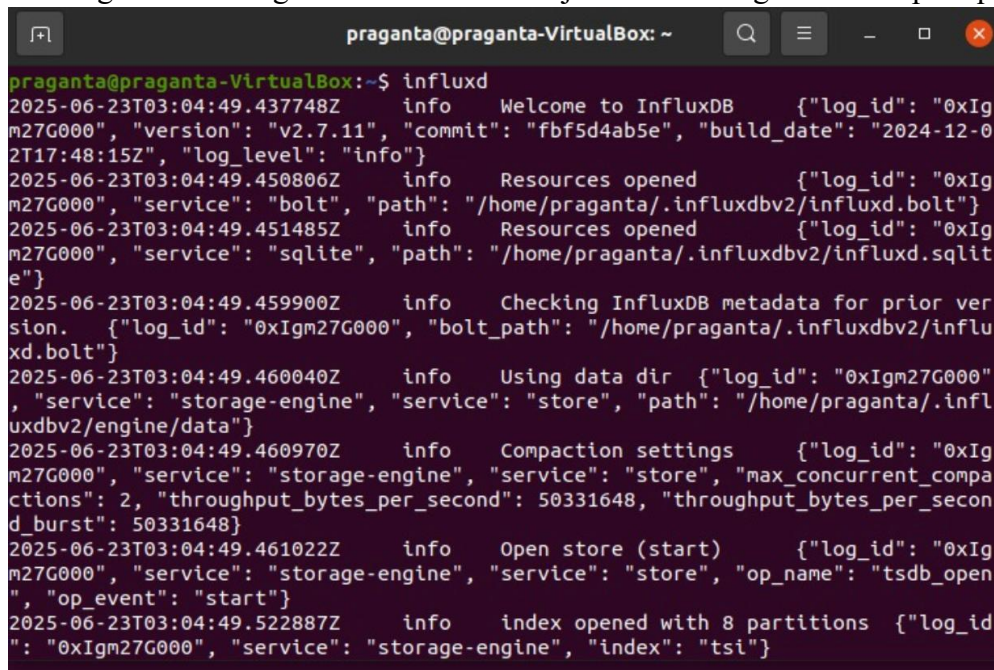
Gambar 1. Ganache

2. Buka new terminal untuk memanggil mytruffle dengan prompt `> cd mytruffle` `>` maka nanti kita akan gunakan contact address mytruffle agar aktif.



Gambar 2. Mytruffle

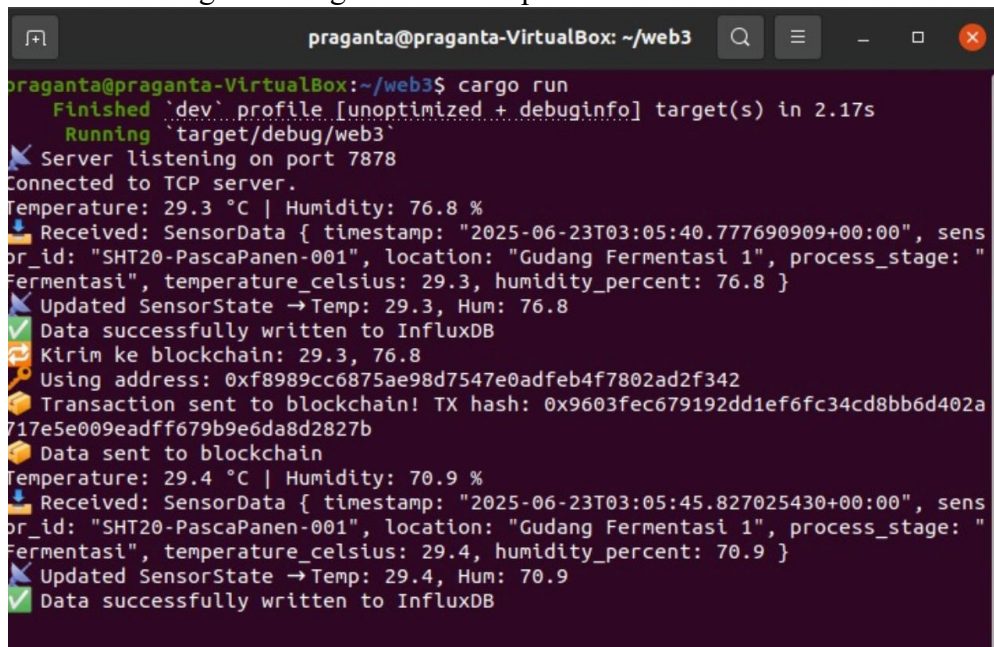
3. Lalu kita new terminal lagi untuk membuka InfluxDb dengan prompt `> influxd`  
> agar nanti mengambil data dan melanjutkan baik ke grafana maupun qt.



```
praganta@praganta-VirtualBox:~$ influxd
2025-06-23T03:04:49.437748Z info Welcome to InfluxDB {"log_id": "0xIgm27G000", "version": "v2.7.11", "commit": "fbf5d4ab5e", "build_date": "2024-12-02T17:48:15Z", "log_level": "info"}
2025-06-23T03:04:49.450806Z info Resources opened {"log_id": "0xIgm27G000", "service": "bolt", "path": "/home/praganta/.influxdbv2/influxd.bolt"}
2025-06-23T03:04:49.451485Z info Resources opened {"log_id": "0xIgm27G000", "service": "sqlite", "path": "/home/praganta/.influxdbv2/influxd.sqlite"}
2025-06-23T03:04:49.459900Z info Checking InfluxDB metadata for prior version. {"log_id": "0xIgm27G000", "bolt_path": "/home/praganta/.influxdbv2/influxd.bolt"}
2025-06-23T03:04:49.460040Z info Using data dir {"log_id": "0xIgm27G000", "service": "storage-engine", "service": "store", "path": "/home/praganta/.influxdbv2/engine/data"}
2025-06-23T03:04:49.460970Z info Compaction settings {"log_id": "0xIgm27G000", "service": "storage-engine", "service": "store", "max_concurrent_compactions": 2, "throughput_bytes_per_second": 50331648, "throughput_bytes_per_second_burst": 50331648}
2025-06-23T03:04:49.461022Z info Open store (start) {"log_id": "0xIgm27G000", "service": "storage-engine", "service": "store", "op_name": "tsdb_open", "op_event": "start"}
2025-06-23T03:04:49.522887Z info index opened with 8 partitions {"log_id": "0xIgm27G000", "service": "storage-engine", "index": "tsi"}
```

Gambar 3. InfluxDb

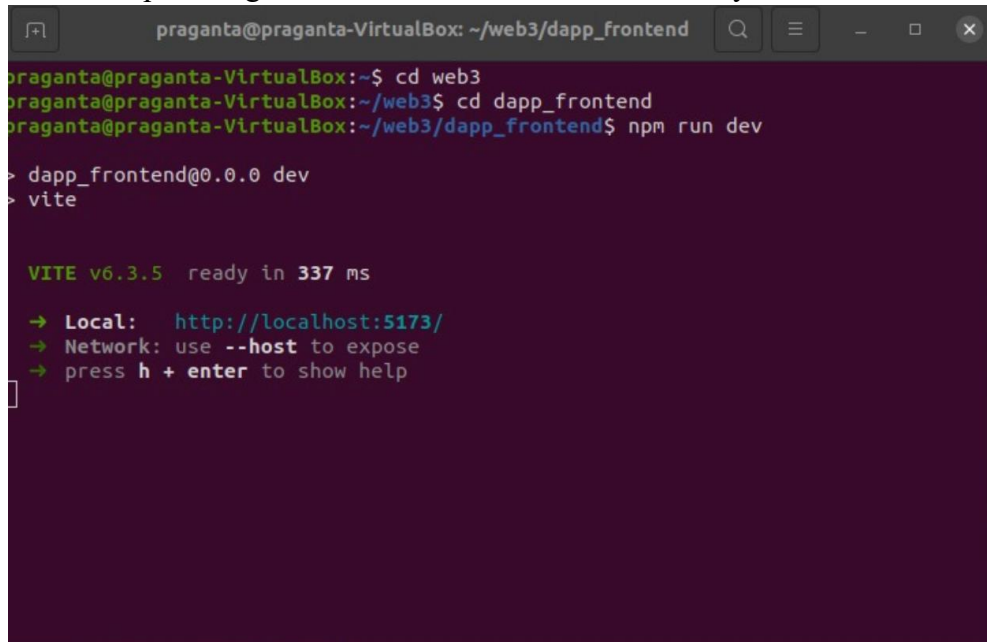
4. Lalu kita new terminal lagi untuk membuka ke web3 dengan prompt `> cd web3`  
> `cargo run` > agar data dari tcp server bisa kita lihat secara real-time.



```
praganta@praganta-VirtualBox:~/web3$ cargo run
Finished `dev` profile [unoptimized + debuginfo] target(s) in 2.17s
Running `target/debug/web3`
Server listening on port 7878
Connected to TCP server.
Temperature: 29.3 °C | Humidity: 76.8 %
Received: SensorData { timestamp: "2025-06-23T03:05:40.777690909+00:00", sensor_id: "SHT20-PascaPanen-001", location: "Gudang Fermentasi 1", process_stage: "Fermentasi", temperature_celsius: 29.3, humidity_percent: 76.8 }
Updated SensorState → Temp: 29.3, Hum: 76.8
Data successfully written to InfluxDB
Kirim ke blockchain: 29.3, 76.8
Using address: 0xf8989cc6875ae98d7547e0adfeb4f7802ad2f342
Transaction sent to blockchain! TX hash: 0x9603fec679192dd1ef6fc34cd8bb6d402a717e5e009eadff679b9e6da8d2827b
Data sent to blockchain
Temperature: 29.4 °C | Humidity: 70.9 %
Received: SensorData { timestamp: "2025-06-23T03:05:45.827025430+00:00", sensor_id: "SHT20-PascaPanen-001", location: "Gudang Fermentasi 1", process_stage: "Fermentasi", temperature_celsius: 29.4, humidity_percent: 70.9 }
Updated SensorState → Temp: 29.4, Hum: 70.9
Data successfully written to InfluxDB
```

Gambar 4. Web3

5. Setelah itu kita membuka new terminal lagi untuk akses ke UI dari web3 nya dengan prompt > cd web3 > cd dapp\_frontend > npm run dev > yang nantinya kita dapat mengakses lokal host untuk melihat hasilnya.



```
praganta@praganta-VirtualBox: ~/web3/dapp_frontend
praganta@praganta-VirtualBox:~$ cd web3
praganta@praganta-VirtualBox:~/web3$ cd dapp_frontend
praganta@praganta-VirtualBox:~/web3/dapp_frontend$ npm run dev

> dapp_frontend@0.0.0 dev
> vite

VITE v6.3.5 ready in 337 ms

➔ Local:   http://localhost:5173/
➔ Network: use --host to expose
➔ press h + enter to show help
```

**Gambar 5.** Tampilan untuk membuka localhost web3

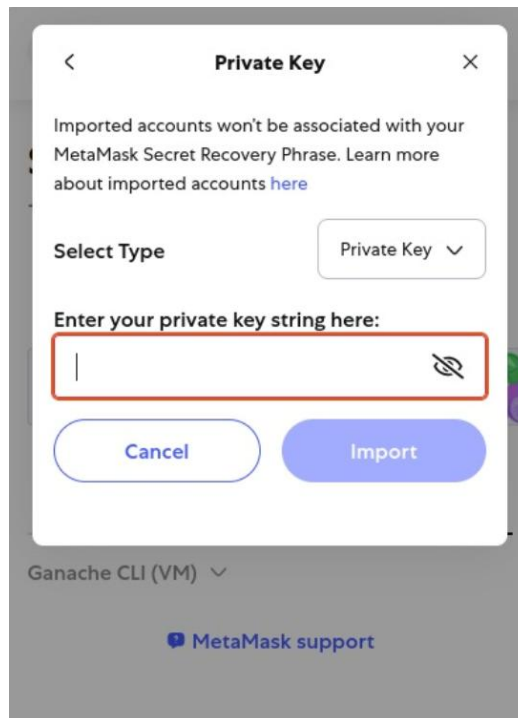
6. Setelah kita buka tampilan UI, kita harus mengubah private key dari metamask agar bisa connect terhubung dengan data kita.



**Gambar 6.** Tampilan UI

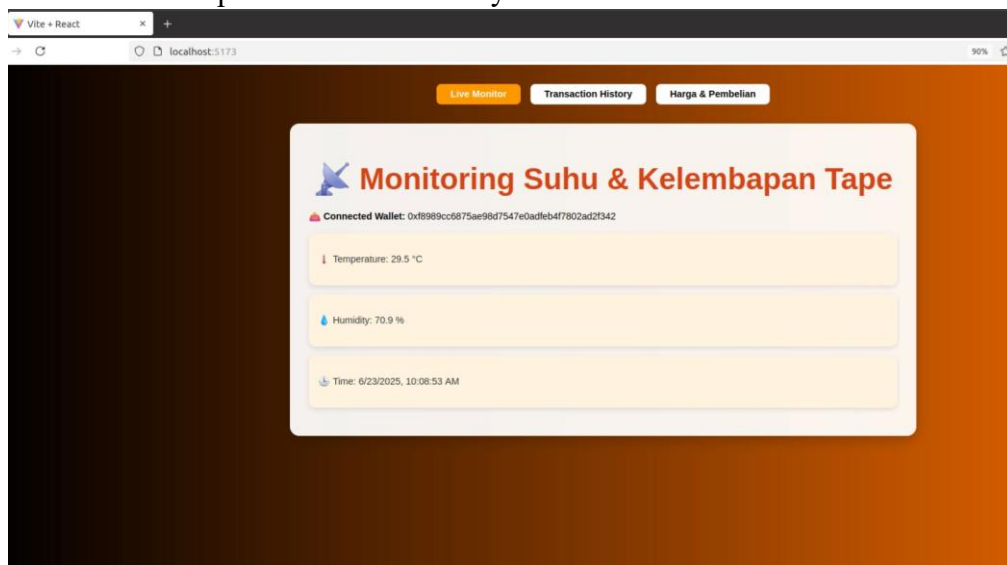


7. Lalu kita ubah private key dari private key ganache awal tadi untuk bisa connect dengan data awal.



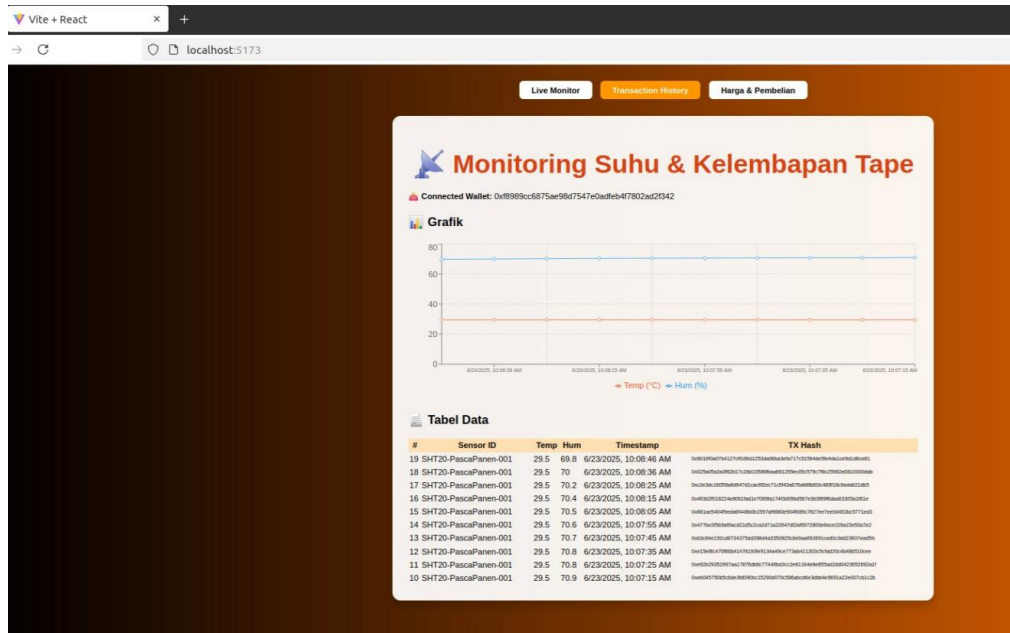
**Gambar 7.** Private Key Metamask

8. Nah setelah private key diganti data akan otomatis connect dan kita bisa melihat Temperature dan Humidity.

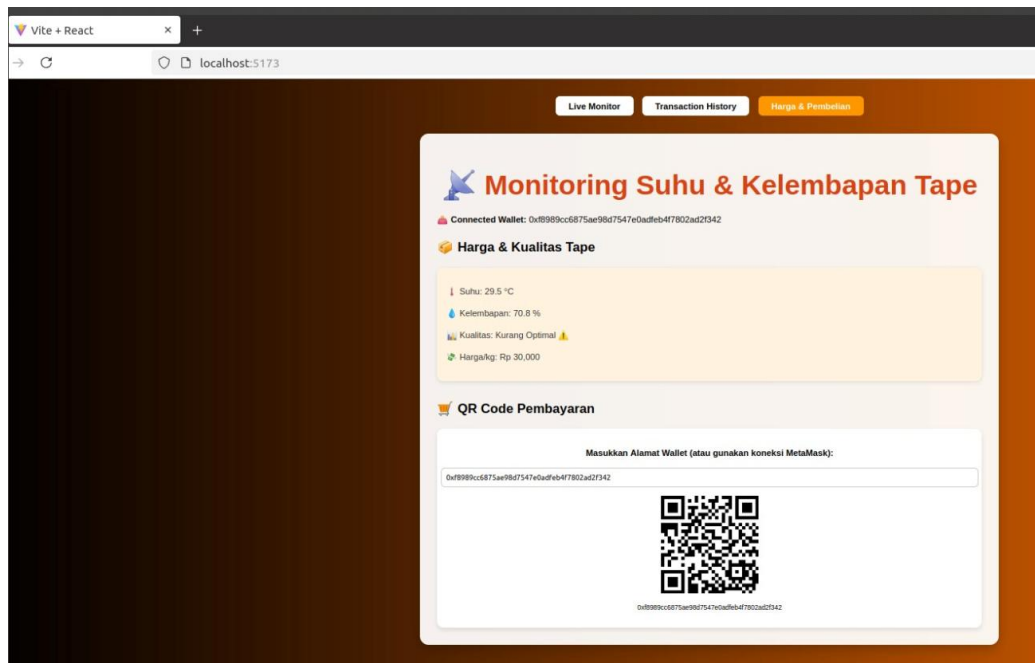


**Gambar 8.** Tampilan Monitoring Suhu dan Kelembapan

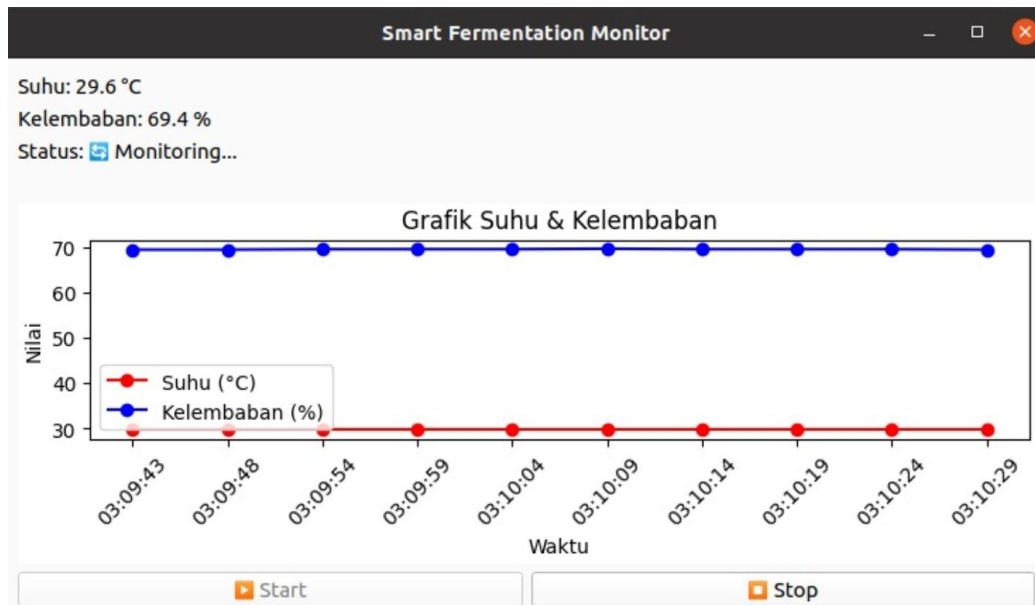
## 4.2 Hasil



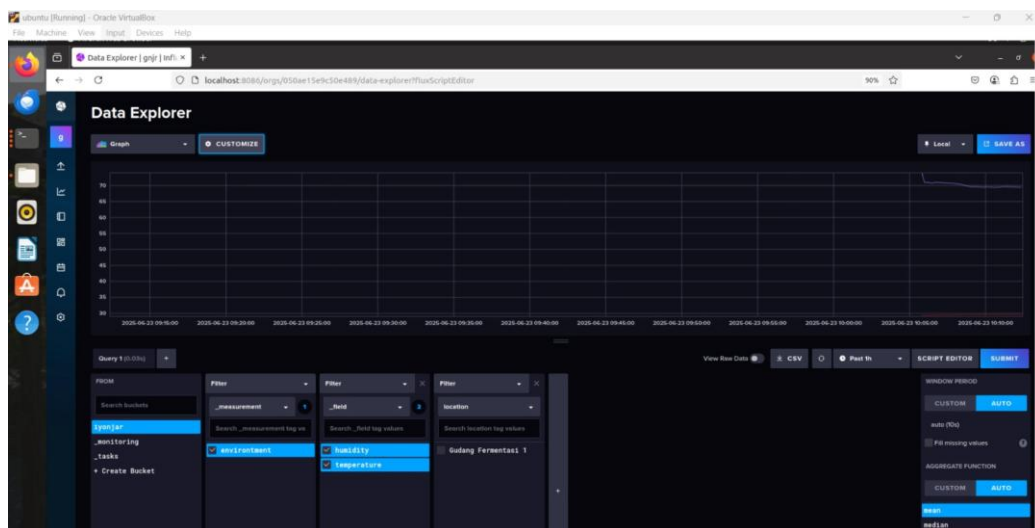
Gambar 9. Tampilan UI Transaction History



Gambar 10. Tampilan UI Harga dan Pembelian



Gambar 11. Tampilan pada Qt



Gambar 12. Tampilan pada Grafana



## Daftar Pustaka

- Priyanto, R., Nugraha, A. B., & Haryanto, M. (2024). Design of Industrial Temperature and Humidity Monitoring System Using SHT20 and Modbus Protocol. *IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, 1–6.
- Sharma, P., Arora, V., & Patel, D. (2023). Reliable Data Acquisition in Rust for Industrial IoT Applications. *IEEE Internet of Things Journal*, 10(5), 4521–4530.
- Sopasakis, P., Zhang, D., & Nia, A. M. (2020). Secure Data Transport and Storage Using TCP and InfluxDB for IoT Systems. *IEEE Transactions on Industrial Informatics*, 16(12), 7702–7710.
- Aggoune, R., & Benratem, D. (2023). Real-Time Visualization of Sensor Data with Grafana in Industrial Environments. *IEEE Access*, 11, 78901–78910.
- Nabijonovich, A., & Najmiddin, K. (2024). Qt-Based GUI Development for Environmental Monitoring Systems. *IEEE Eurasian Conference on Educational and Scientific Innovations (EESI)*, 112–117.
- Doan, T. (n.d.). *Temperature and humidity transmitter SHT20 sensor Modbus RS485*. Scribd. <https://www.scribd.com/document/486890382/Temperature-and-Humidity-Transmitter-SHT20-Sensor-Modbus-RS485>
- Scribd. (2023). *XY-MD02 RS485 Temperature and Humidity Sensor Manual*. Scribd. <https://www.scribd.com/document/555348739/XY-MD02-RS485-Temperature-and-Humidity-Sensor-Manual>
- gottliebfs. (n.d.). *XY-MD02 information: Modbus RS485 RTU temperature and humidity sensor documentation*. Scribd. <https://www.scribd.com/document/481896668/XY-MD02-information>
- MOHAMED ALI OTHMAN. (n.d.). *SHT4x (SHT40) Datasheet: High-accuracy humidity & temperature sensor platform*. Scribd. <https://www.scribd.com/document/575893785/SHT40-Datasheet>