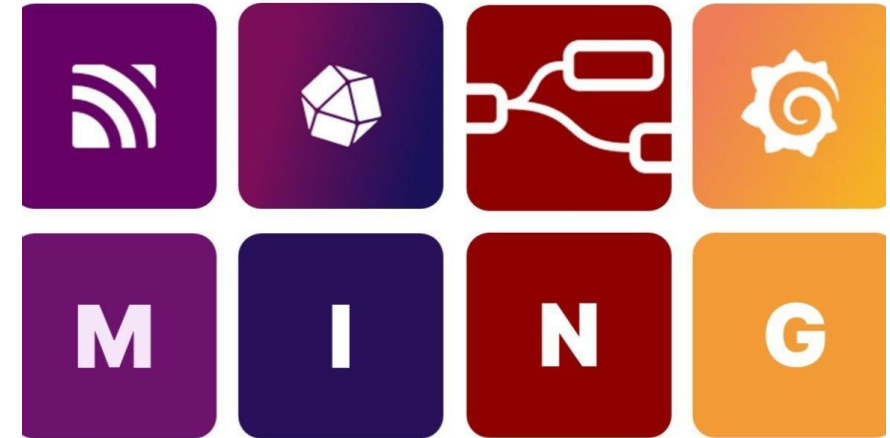


MING STACK IOT (MQTT – INFLUXDB – NODERED – GRAFANA)

OUTLINE

1. Pengenalan **MING Stack IoT**
2. MQTT Sebagai Protokol Komunikasi
3. Node Red Sebagai *Backend Server*
4. InfluxDB Sebagai *Time Series Database*
5. Grafana Sebagai *User Interface*
6. **Demo Ming Stack IoT (Hands on)**



<https://grafana.com/api/dashboards/12921/images/8829/image>



TAK KENAL MAKA **MARI KENALAN**



S1 Kimia Universitas Hasanuddin

- *Laboratory Officer* **Bosowa School Makassar**
- *Owner* **Rizky Project**
- *Trainer IoT & Data Science* **Indobot Academy**
- *Pengurus Komunitas* **ArduMeka**



085332257589



Rizky Dermawan



Rizky Dermawan



Rizky Dermawan



Rizky Dermawan



<https://github.com/rizkydermawan1992>



Rizky Dermawan, S.Si.



*Backendnya pake Nodered
terus integrasi ke InfluxDB.
On premise aja sih di Raspi
termasuk broker MQTTnya*



*Proyek IoT Lu
stacknya apa aja
bro?*

**Lu yang cuma taunya kirim data
sensor ke BLYNK**



KESENJANGAN ARSITEKTUR

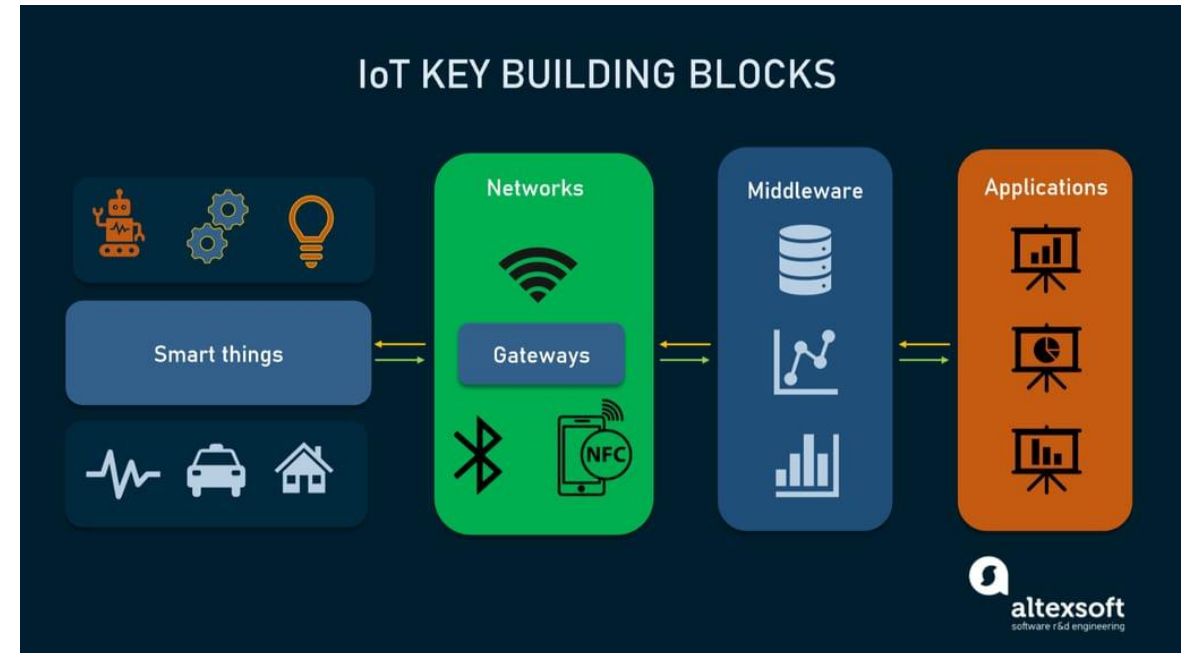


PENGENALAN MING STACK IOT



ARSITEKTUR IOT

- **PERCEPTION LAYER** – Mengumpulkan data dari lingkungan fisik
- **TRANSPORT LAYER** – Mengirimkan data dari perangkat ke pusat pemrosesan data
- **PROCESSING LAYER** – Mengolah, menyimpan, dan menganalisis Data dari perangkat
- **APPLICATION LAYER** - Menyediakan layanan dan informasi kepada pengguna berdasarkan data yang telah diproses

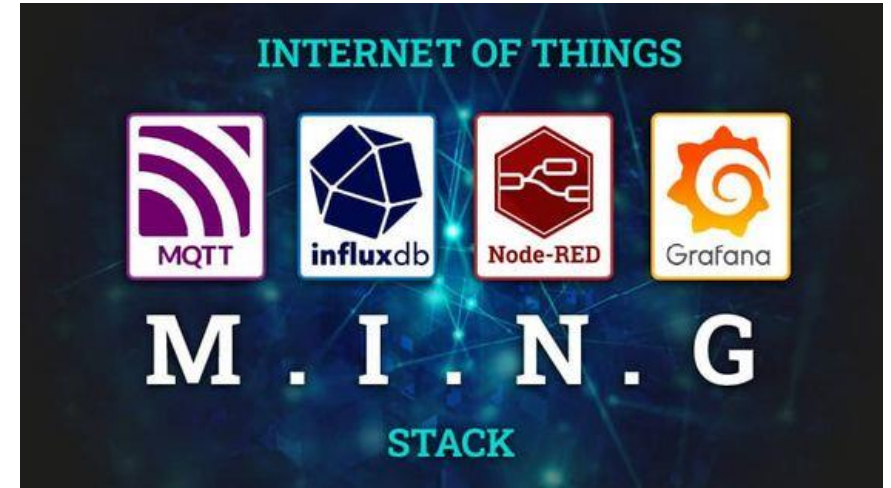


<https://www.altexsoft.com/blog/iot-architecture-layers-components/>

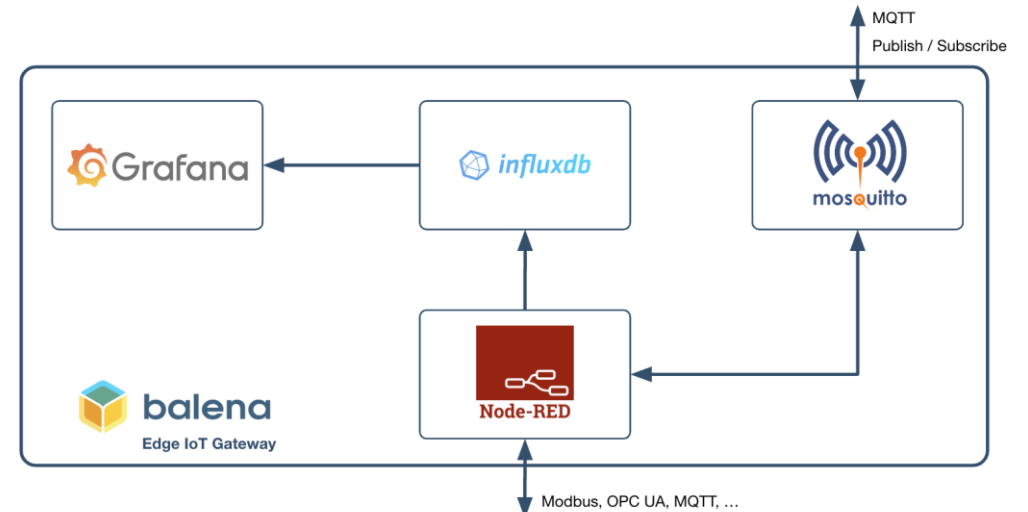


APA ITU MING STACK ?

- Merupakan gabungan 4 teknologi *open-source* yang bisa digunakan di perangkat edge untuk mengumpulkan, mengolah, dan menampilkan data langsung dari perangkat IoT, seperti PLC atau sensor LoRaWAN yang mengirim data ke *LoRa Network Server* (LNS)
- Pertama kali diinisiasi oleh Tim Balena pada tahun 2019 yang terinspirasi dari LAMP stack untuk Web Development
- Menyediakan solusi lengkap *end-to-end* mulai dari pengumpulan data sampai visualisasi
- Cocok untuk proyek *edge computing* di lapangan tanpa harus selalu bergantung pada cloud



<https://flowfuse.com/blog/2023/02/ming-blog/>



<https://blog.balena.io/ming-stack-mqtt-influxdb-nodered-grafanaa-balena/>



EDGE COMPUTING

- Merupakan paradigma komputasi dimana pemrosesan data dilakukan sedekat mungkin dengan sumber data.
- Respon lebih cepat (*low latency*) karena data tidak perlu dikirimkan *cloud server*
- Meminimalisir *bandwidth* karena hanya data yang penting saja yang dikirimkan ke cloud server
- Keamanan data dan privasi terjaga karena data sensitif dikelola secara lokal
- Tidak bergantung pada koneksi internet karena pemrosesan data dilakukan secara lokal

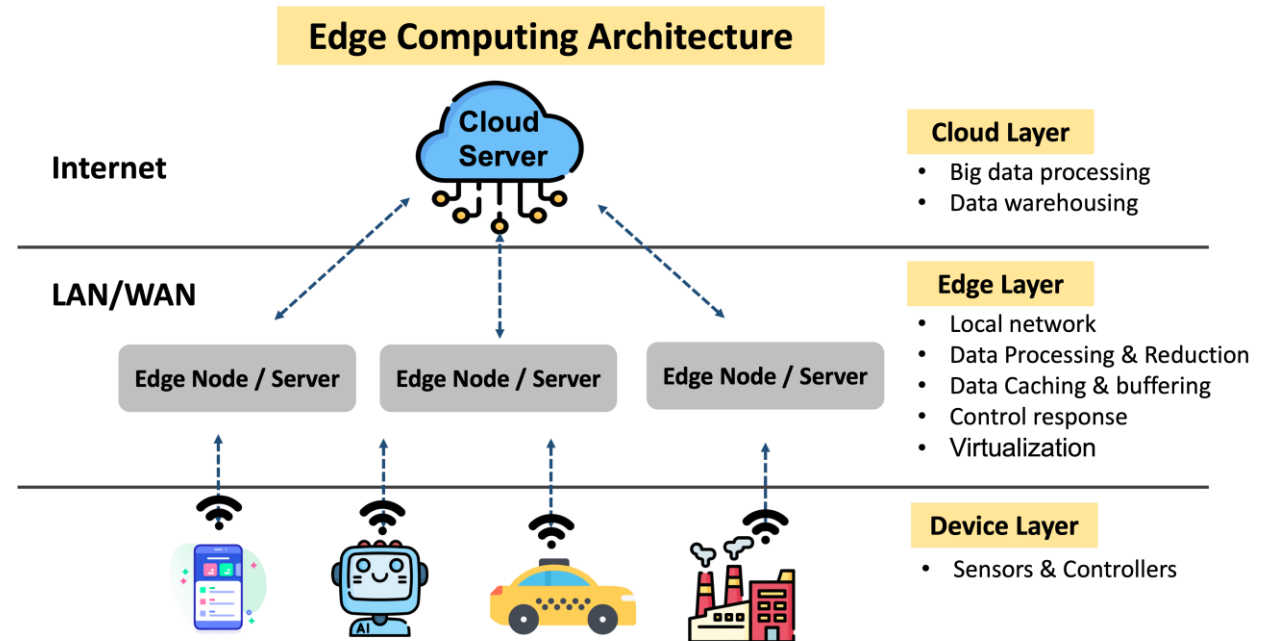


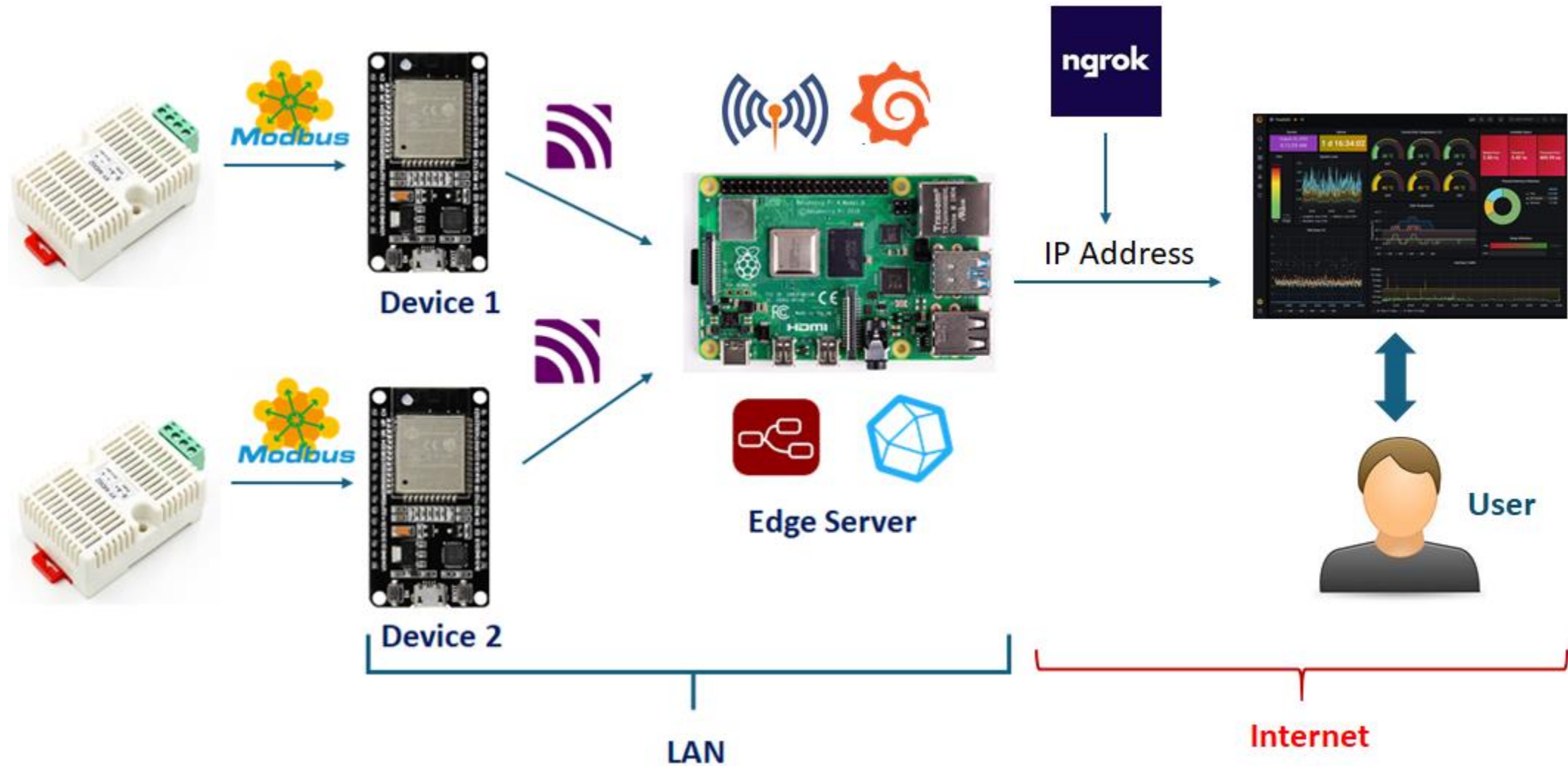
Figure : Edge computing architecture overview

Source : The research team

<https://www.fsp-group.com/en/knowledge-app-42.html>



DIAGRAM IMPLEMENTASI MING STACK



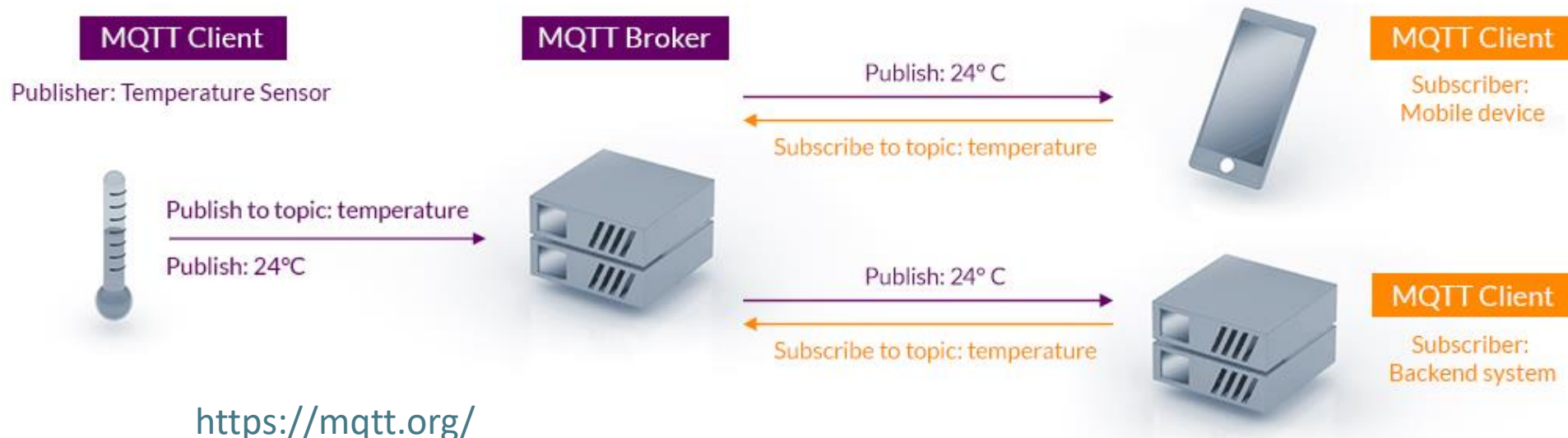


MQTT SEBAGAI PROTOKOL KOMUNIKASI



PENGENALAN MQTT

- Merupakan protokol ringan untuk komunikasi *machine to machine* yang berjalan di atas TCP/IP
- Komunikasi data menggunakan model **publish-subscribe**
- Perangkat pengirim pesan disebut **publisher** dan penerima pesan disebut **subscriber** yang keduanya harus terhubung pada satu **broker** dan **topik** yang sama
- **Broker** merupakan server pusat dalam mengorganisir pesan antara publisher dan subscriber
- **Topik** adalah jalur komunikasi untuk merutekan pesan antara publisher dan subscriber





ECLIPSE MOSQUITTO

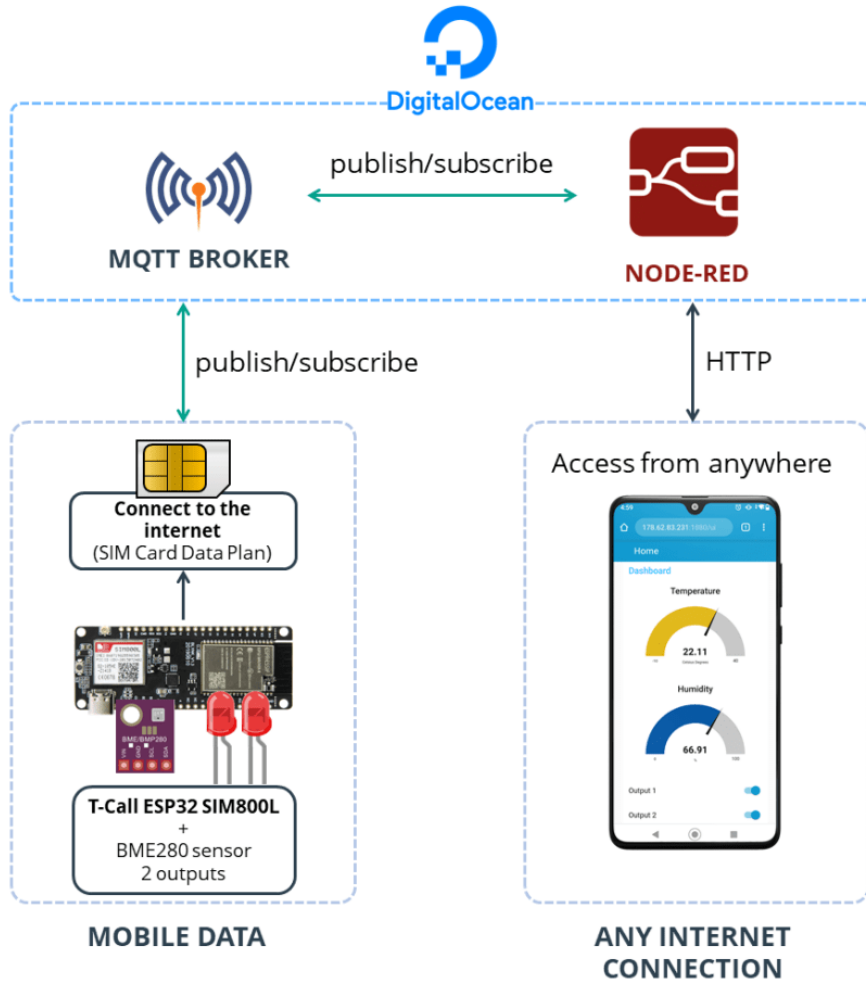
- **Eclipse Mosquitto** merupakan broker pesan MQTT yang ringan dan bersifat *open source*
- Mendukung beberapa versi MQTT 3.1, 3.1.1, 5.0
- Cocok digunakan untuk semua jenis perangkat mulai dari SBC berdaya rendah sampai *fully server*
- Kompatibel dengan berbagai bahasa pemrograman seperti C, Python, dan Javascript
- Dapat dijalankan secara **on premise** (localhost) atau menggunakan broker **publik** (cloud)

Port	Protokol	Keterangan
1883	MQTT (tanpa TLS)	Koneksi biasa (unencrypted)
8883	MQTT (dengan TLS)	Koneksi aman (encrypted)
8080	MQTT over WebSockets	Tanpa TLS
8081	MQTT over WebSockets (TLS)	Aman via browser

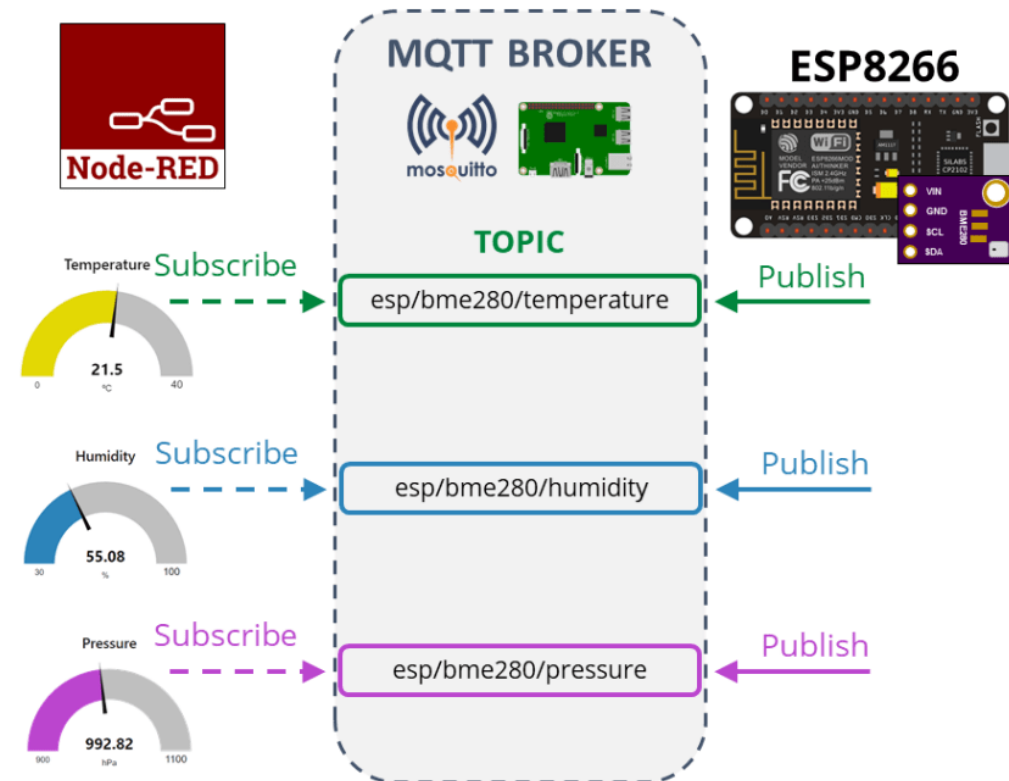


<https://mosquitto.org/>

TOPOLOGI PUBLISH-SUBSCRIBE



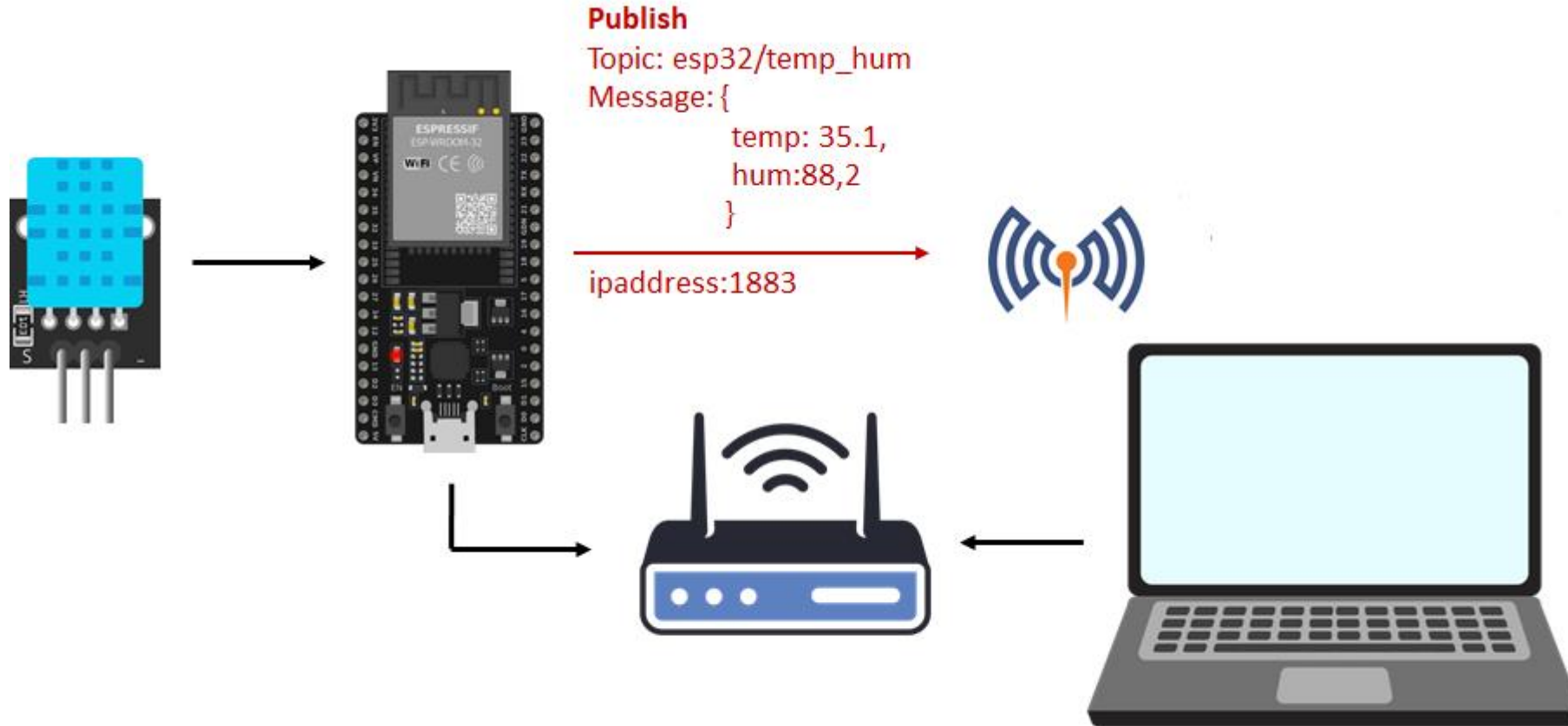
<https://randomnerdtutorials.com/esp32-cloud-mqtt-broker-sim800l/#more-99705>



<https://randomnerdtutorials.com/esp8266-nodemcu-mqtt-publish-bme280-arduino/#more-96350>



CONTOH PENERAPAN MQTT PADA ARSITEKTUR IOT

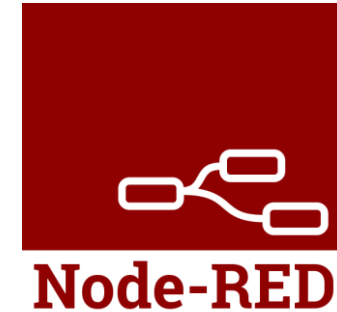




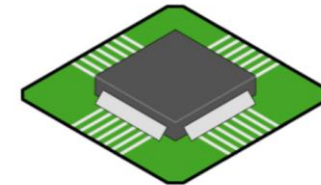
NODERED SEBAGAI *BACKEND SERVER*

DASAR NODE RED

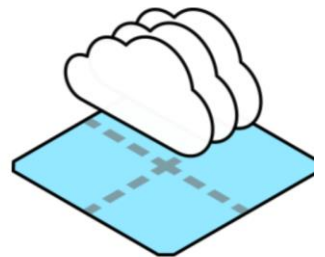
- **Node-RED** adalah *flow-based development tool* berbasis web untuk memprogram aplikasi menggunakan *drag-and-drop interface*.
- Dibangun di atas **Node.js**, sehingga mendukung ekosistem JavaScript.
- Cocok untuk integrasi antara **perangkat keras** (seperti Arduino, ESP32), API, dan layanan online (seperti MQTT, HTTP, dll).
- Memiliki **antarmuka visual berbasis browser** yang memungkinkan pengguna membangun logika program tanpa perlu menulis kode secara penuh.
- Bisa dijalankan di berbagai perangkat, mulai dari PC, Raspberry Pi, hingga server cloud atau kontainer Docker.



Run locally



On a device



In the cloud

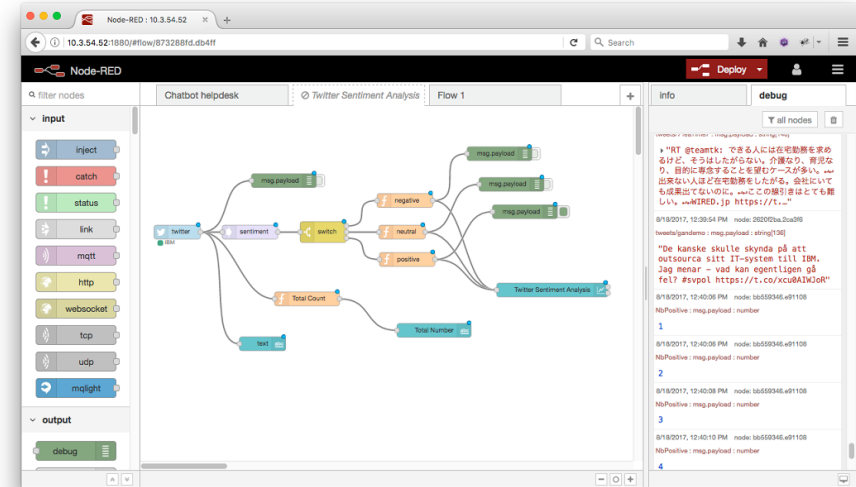
<https://nodered.org/>



STRUKTUR UTAMA NODERED

1. Flow Editor (Editor Visual)

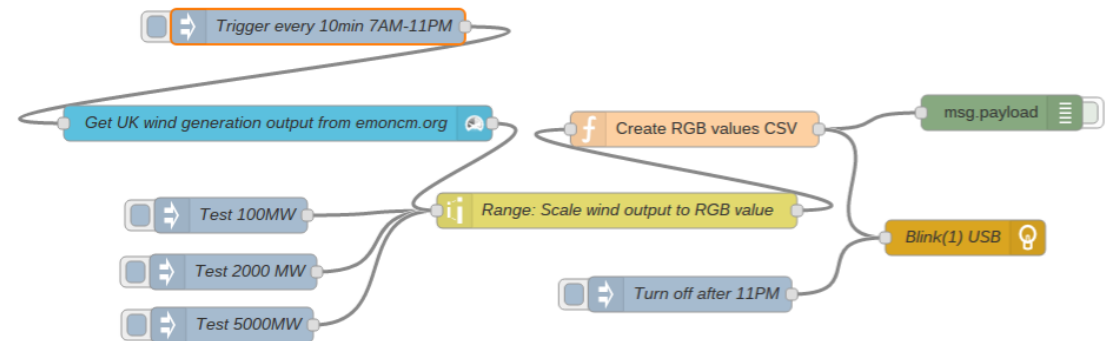
- Merupakan **antarmuka utama** Node-RED berbasis web.
- Digunakan untuk menyusun, menghubungkan, dan mengatur node secara visual dengan cara drag-and-drop.
- Komponen dalam editor: sidebar, canvas, toolbar, debug



<https://developer.ibm.com/tutorials/i-running-node-red/>

2. Flow

- Flow adalah **alur logika atau rangkaian proses** yang terdiri dari node-node saling terhubung.
- Setiap flow bisa dianggap sebagai satu fungsi atau skenario dalam aplikasi
- Flow dapat di-export dan import sebagai **file .json**



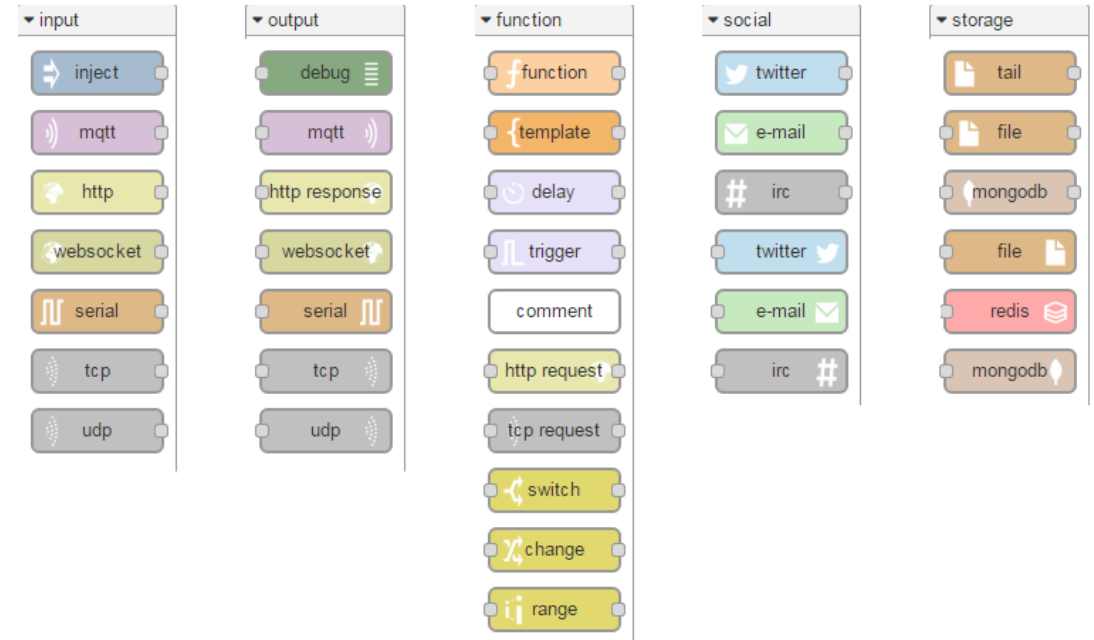
<https://blog.openenergymonitor.org/2015/11/ambient-wind-energy-indicator-using/>



STRUKTUR UTAMA NODERED

3. Node

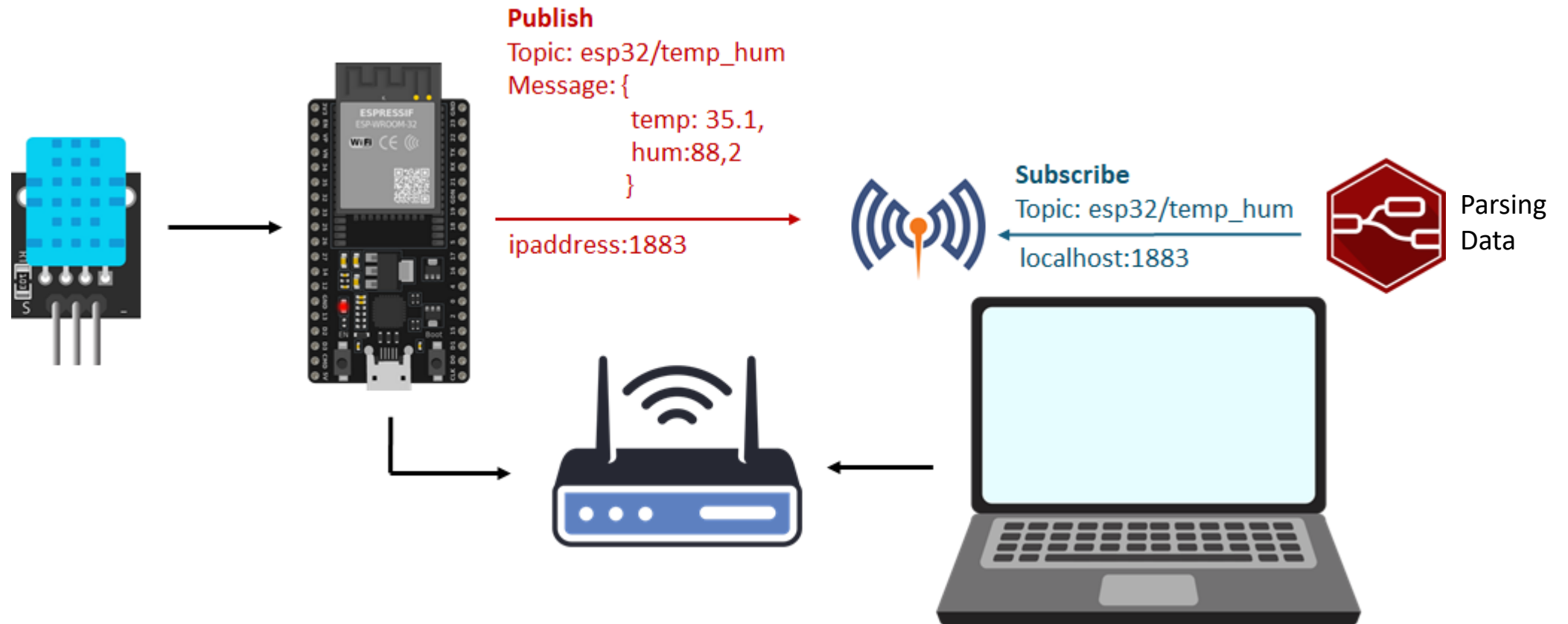
- Node adalah **komponen fungsional utama** di Node-RED yang digunakan untuk **menerima, memproses, dan mengirim data**.
- Setiap node memiliki tugas spesifik, misalnya membaca input, mengolah logika, atau menampilkan output.
- **Input Node**: digunakan untuk memasukkan data ke dalam flow
- **Processing Node**: digunakan untuk mengolah dan memfilter data sebelum masuk ke output
- **Output Node**: digunakan untuk mengirimkan hasil akhir dari flow ke antarmuka



<https://www.cloudfoundry.org/blog/100-day-challenge-018-running-node-red-cloud-foundry/>



CONTOH PENERAPAN NODERED PADA ARSITEKTUR IOT

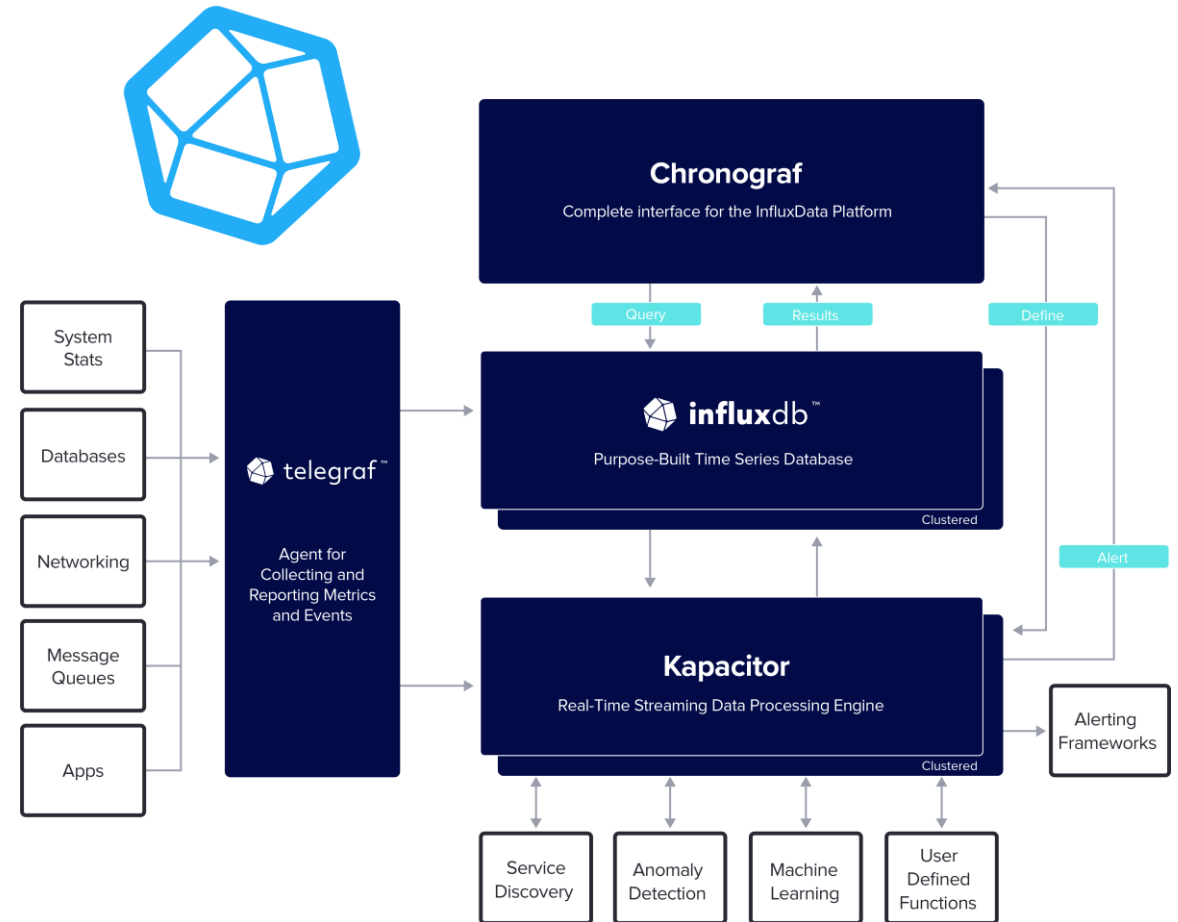




INFLUXDB SEBAGAI *TIME SERIES DATABASE*

PENGENALAN INFLUXDB

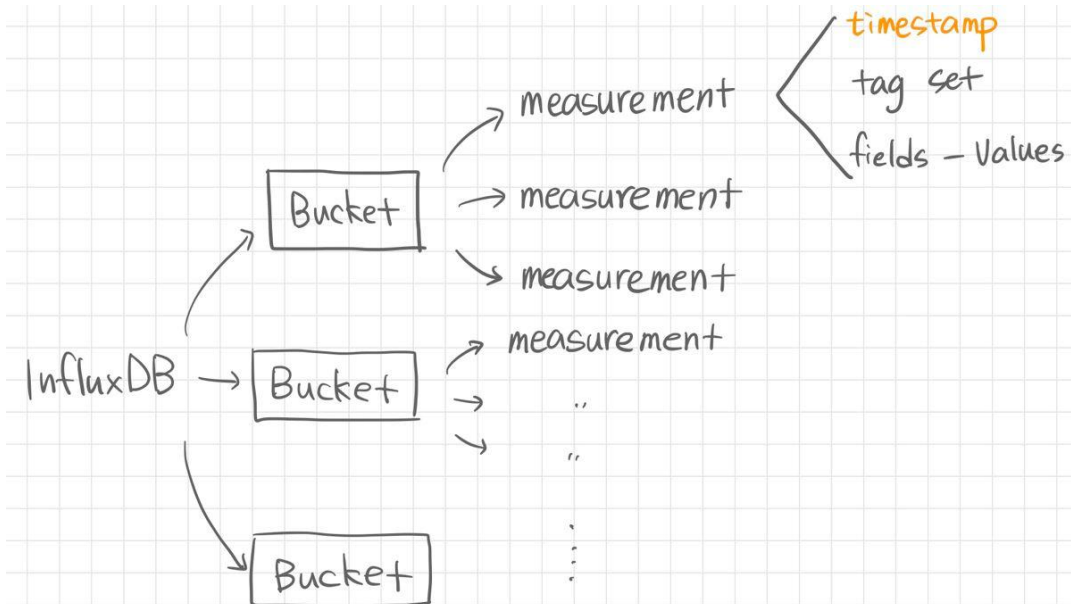
- InfluxDB adalah database khusus untuk menyimpan data berdasarkan waktu (*time series*).
- Dirancang untuk menangani banyak data dengan cepat dan efisien.
- Cocok untuk data sensor, pemantauan sistem (DevOps), dan analisis real-time.
- Bisa diatur agar data lama terhapus otomatis untuk menghemat penyimpanan.
- Memiliki bahasa kueri mirip SQL untuk mengambil dan menganalisis data.



<https://www.influxdata.com/time-series-platform/>



SKEMA DATA INFLUXDB



<https://velog.io/@jee-9/InfluxDB-2-Structure-Data-elements>

Bucket: data_iot

time	device	location	temp	hum
2025-05-09T00:00:00Z	device_1	operator_room	28.5	65
2025-05-09T00:00:00Z	device_2	server_room	30.1	60
2025-05-09T00:05:00Z	device_1	operator_room	28.7	64
2025-05-09T00:05:00Z	device_2	server_room	30.3	59
2025-05-09T00:10:00Z	device_1	operator_room	29.0	63
2025-05-09T00:10:00Z	device_2	server_room	30.5	58

Measurement: data_dht11

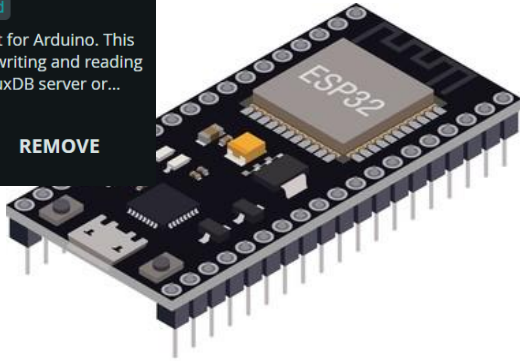
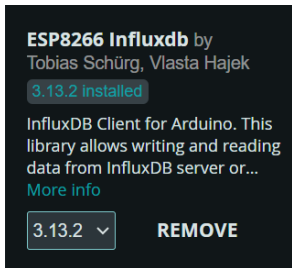


Tag sifatnya opsional namun berperan untuk mempercepat proses query data



MEKANISME PENYIMPANAN DATA INFLUXDB

Library Client



PAYLOAD

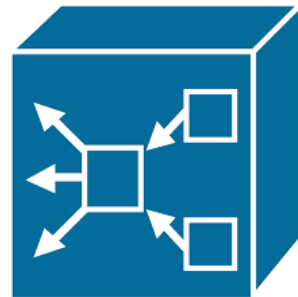


measurement, tags, fields

url, token, org, bucket



METADATA



GATEWAY

PAYLOAD



measurement, tags, fields

url, token, org, bucket

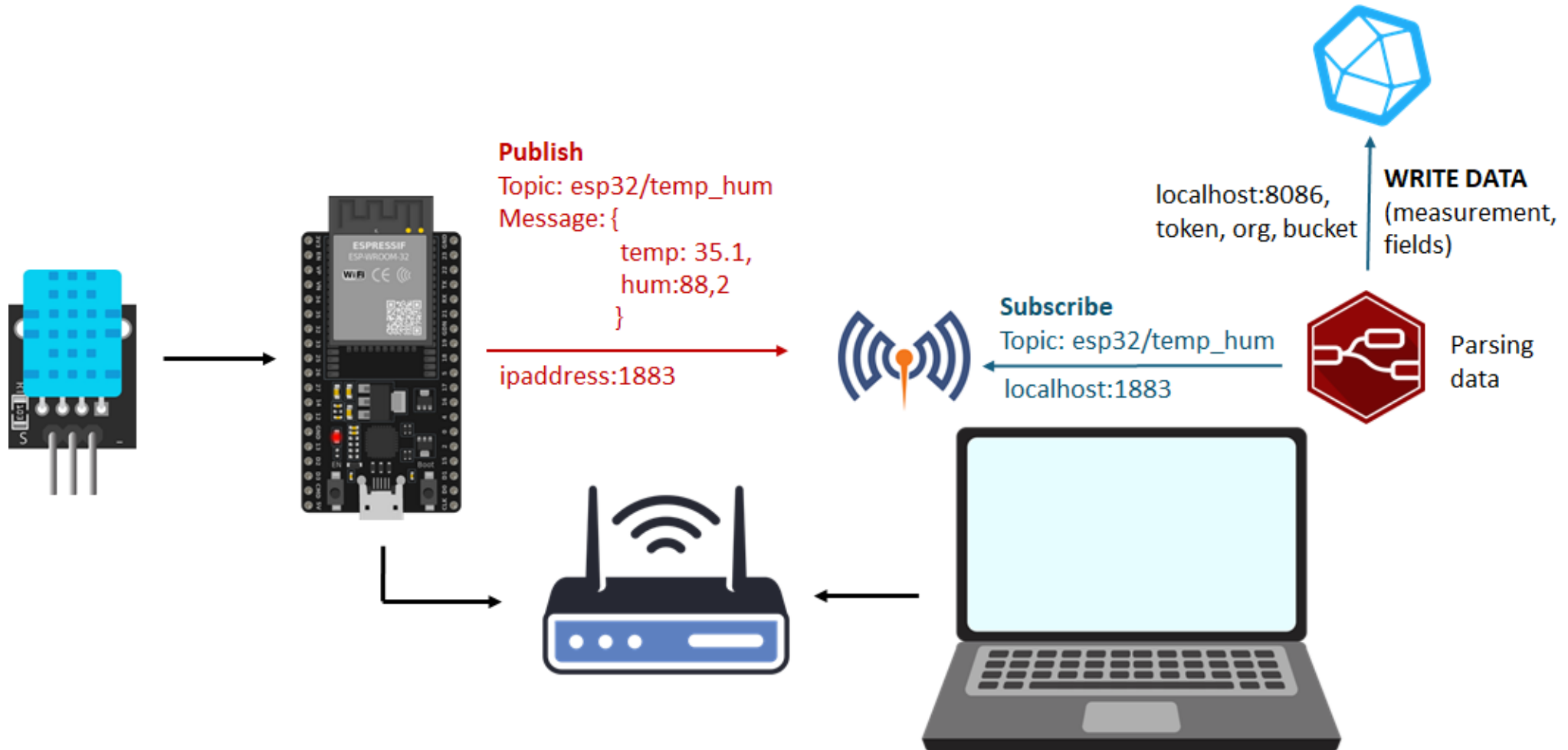


METADATA





CONTOH PENERAPAN INFLUXDB PADA ARSITEKTUR IOT





GRAFANA SEBAGAI *USER INTERFACE*

PENGENALAN GRAFANA

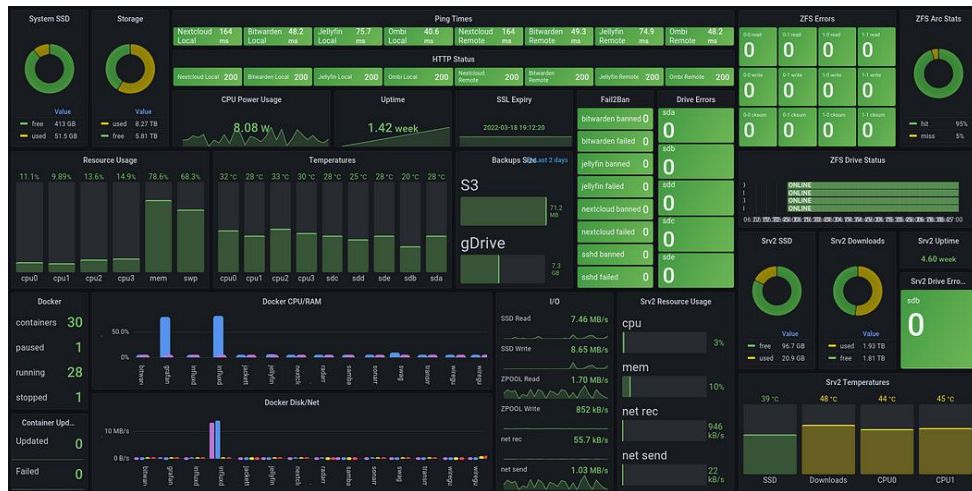
- **Grafana** merupakan tools untuk **visualisasi dan analisis data**.
- Menampilkan data dalam bentuk **grafik interaktif** yang bisa disesuaikan.
- Pengambilan data dari berbagai **data source** (database, IoT, cloud, dll).
- Tampilan dibagi menjadi **panel**, yang disusun menjadi **dashboard**.
- Mendukung **notifikasi otomatis** melalui email dan lainnya.
- Mendukung fitur **alert** untuk mendeteksi perilaku data yang tidak normal.
- Fitur **annotate** memungkinkan user memberi **catatan langsung di grafik**.
- Bersifat **open-source** dan dikembangkan oleh komunitas aktif.



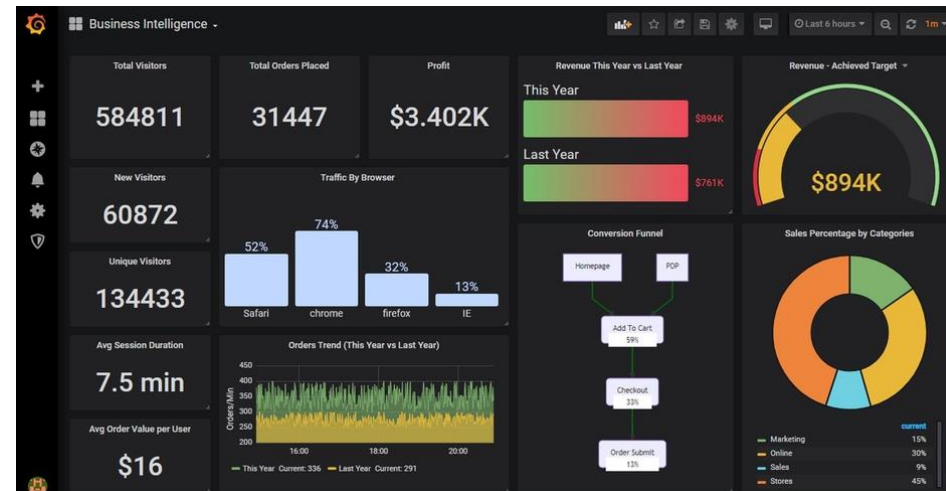
<https://grafana.com/blog/2020/09/04/inside-grafana-labs-more-workspaces-revealed/>

PENGUNAAN GRAFANA DALAM MONITORING

- **Monitoring Infrastruktur:** memantau kesehatan server, database, dan jaringan.
- **Monitoring Aplikasi:** memantau performa aplikasi, seperti waktu respon, error, dan trafik.
- **Monitoring Log:** menganalisis data log dan menemukan pola atau keanehan.
- **Monitoring Perangkat IoT:** memantau data dari sensor suhu, kelembaban, gerak, dll.
- **Monitoring Data Bisnis:** memantau angka penjualan, pendapatan, dan kepuasan pelanggan.



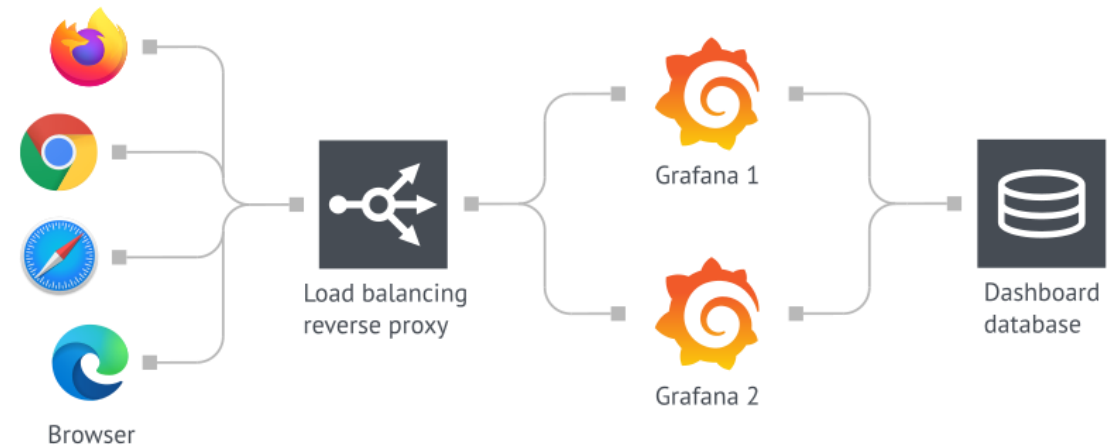
<https://medium.com/@james.ralph8555/setting-up-grafana-with-influxdb-for-server-monitoring-7b16c1d0ba0c>



<https://medium.com/@MetricFire/grafana-vs-powerbi-using-grafana-for-your-business-metrics-360bfe70f06c>

ARSITEKTUR GRAFANA

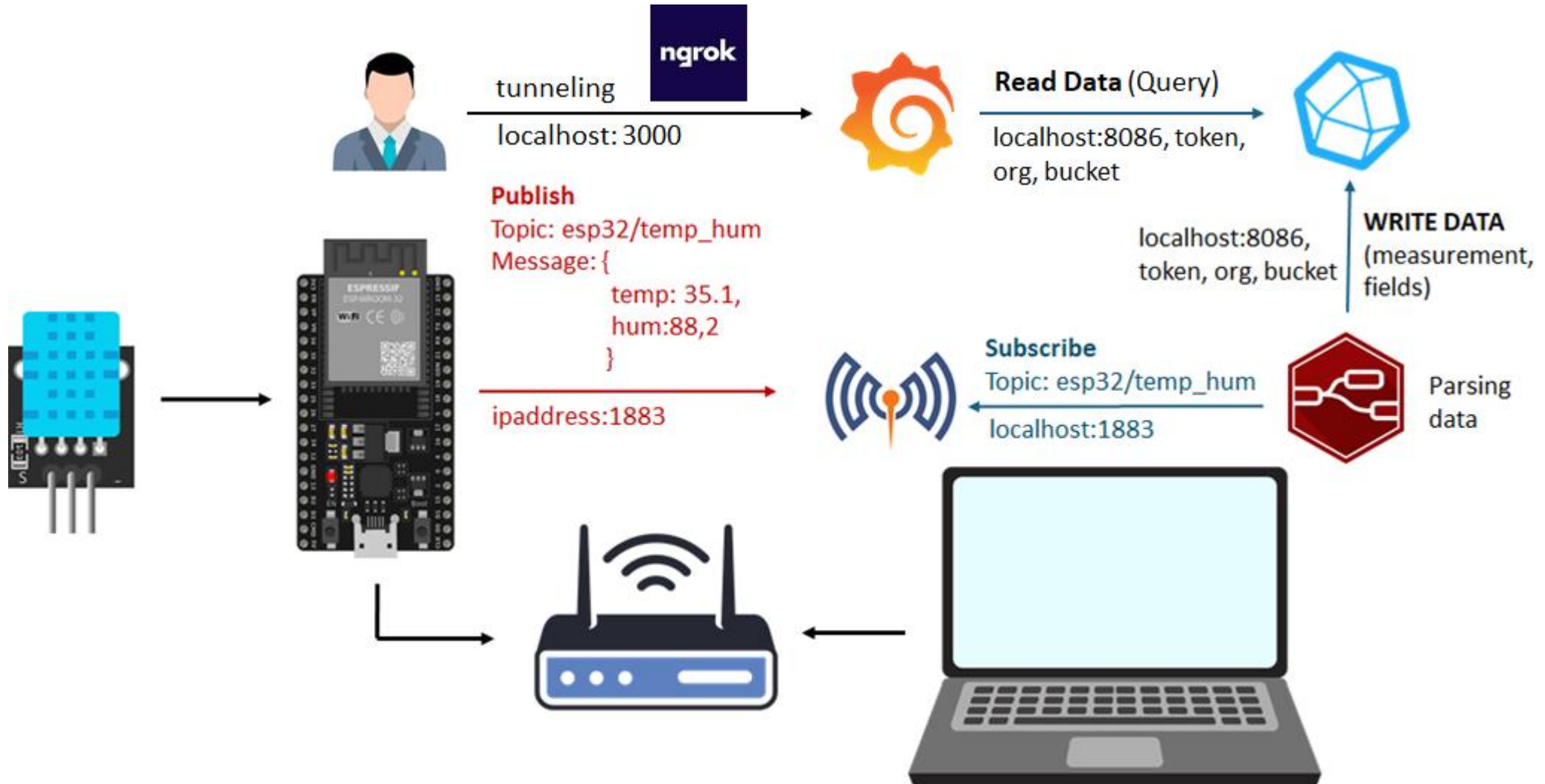
- Beberapa server grafana membentuk **node cluster**
- *Traffic* disebar ke beberapa server melalui **load balancer**
- Setiap node server aktif menangani permintaan
- Jika ada node yang gagal maka load balancer mengalihkan traffic ke node yang masih aktif
- **Shared Hostname** diterapkan agar akses ke layanan Grafana hanya melalui satu alamat saja
- Grafana menggunakan **database sqlite 3** untuk menyimpan data user, dashboard dan data lainnya
- Arsitektur ini untuk menjaga **High Availability** dari server



<https://grafana.com/docs/grafana/latest/setup-grafana/set-up-for-high-availability/>



CONTOH PENERAPAN GRAFANA (+ NGROK) PADA ARSITEKTUR IOT





DEMO MING STACK IOT (HANDS ON)

ALUR KERJA SISTEM MONITORING ESP32-DHT11

1. Sensor DHT11 melakukan *sensing* di lingkungan lalu mengirim sinyal ke ESP32
2. ESP32 memproses sinyal menjadi data suhu dan kelembaban lalu mempublish ke broker MQTT (Mosquitto)
3. Mosquitto meneruskan data ke Nodered setelah Nodered sebagai client melakukan subscribe
4. Nodered melakukan parsing data lalu melakukan write data ke Influxdb
5. Influxdb melakukan stream data ke server Grafana setelah *request access* Grafana diizinkan
6. Grafana memvisualisasikan data ke user melalui dashboard

