

SKRIPSI

STUDI KOMPARASI *DATABASE* MYSQL MONGODB DAN INFLUXDB
SEBAGAI DATA *STORAGE* SENSOR BERBASIS IOT

Diajukan Sebagai Salah Satu Persyaratan Untuk Memperoleh Gelar Sarjana (S1)

Program Studi Teknik Elektro Fakultas Teknik Universitas Tidar



Disusun Oleh:

Nama : Ahmad Robih Rizqi

NPM : 1810501077

PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS TIDAR

2023

HALAMAN PENGESAHAN

Proposal skripsi ini diajukan sebagai salah satu syarat memperoleh gelar Sarjana Teknik Jurusan Teknik Elektro, Fakultas Teknik, Universitas Tidar dengan judul:

STUDI KOMPARASI *DATABASE* MYSQL MONGODB DAN INFLUXDB SEBAGAI DATA *STORAGE* SENSOR BERBASIS IOT

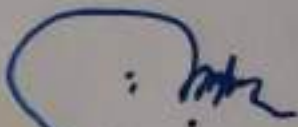
Disusun Oleh:

Nama : Ahmad Robih Rizqi

NPM : 1810501077

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I,

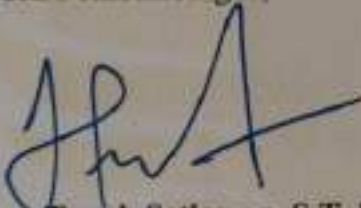


Ir. Ibrahim Nawawi, S.T., M.T., IPM.

NIP. 197303282021211004

Tanggal: 27 Juni

Dosen Pembimbing II,



Hery Teguh Setiawan, S.T., M.Eng.

NIP. 198701092019031005

Tanggal: 26 Juni

Mengetahui :

Dekan Fakultas Teknik

Universitas Tidar



Dr. Ir. Gito Sugiyanto, S.T., M.T., IPM., ASEAN.Eng.

NIP. 198002152002121003

LEMBAR PENGUJI

Skripsi yang berjudul:

STUDI KOMPARASI *DATABASE* MYSQL MONGODB DAN INFLUXDB SEBAGAI DATA *STORAGE* SENSOR BERBASIS IOT

Disusun Oleh:

Nama : Ahmad Robih Rizqi

NPM : 1810501077

Telah berhasil dipertahankan di depan Dewan Penguji dalam ujian pendadaran skripsi dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik Jurusan Teknik Elektro Fakultas Teknik Universitas Tidar.

Magelang,

2023

Dihadapan Penguji:

1. Ir. Ibrahim Nawawi, S.T., M.T., IPM.

NIP. 197303282021211004

2. Hery Teguh Setiawan, S.T., M.Eng.

NIP. 198804292019031005

3. Ir. Bagus Fatkhurrozi, S.T., M.T., IPM.

NIP. 1195909281991031001

Mengetahui,

Dekan Fakultas Teknik

Universitas Tidar

Dr. Ir. Gito Sugiyanto, S.T., M.T., IPM, ASEAN, Eng.

NIP. 198002152002121003

LEMBAR PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Ahmad Robih Rizqi

NPM : 1810501077

Jurusan : Teknik Elektro

Fakultas : Teknik

Menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

STUDI KOMPARASI DATABASE MYSQL MONGODB DAN INFLUXDB SEBAGAI DATA STORAGE SENSOR BERBASIS IOT

Benar-benar hasil karya sendiri, kecuali dalam pengutipan substansi disebutkan sumbernya dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan atau plagiat. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi. Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapatkan sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Magelang, 23 Juni 2023



Ahmad Robih Rizqi

NPM.1810501077

KATA PENGANTAR

Alhamdulillahirobbilalamin. Puji syukur kami panjatkan ke hadirat Allah SWT, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi ini. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana di Jurusan Teknik Elektro. Pada kesempatan ini, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan, bantuan, dan motivasi selama proses penulisan skripsi ini. Ucapan terima kasih ini ditujukan kepada:

1. Rektor Universitas Tidar, Prof. Dr. Sugiyarto, M.Si.
2. Dekan Fakultas Teknik, Dr. Ir. Gito Sugiyanto, S.T., M.T., IPM., ASEAN.Eng.
3. Kepala Jurusan Teknik Elektro dan selaku pembimbing akademik yang telah memberikan waktu, saran dan masukan selama tugas akhir, Ir. Deria Pravitasari, S.T., M.Eng., IPM.
4. Keluarga dan terutama orang tua tercinta yaitu Ibu Sriyatun dan Bapak Ismiyadi, yang selalu memberikan dukungan, doa, dan motivasi selama proses penulisan skripsi ini. Doa dan dorongan yang diberikan sangat berarti bagi penulis dalam menghadapi setiap tantangan dalam penelitian ini.
5. Pembimbing Skripsi saya, Ir. Ibrahim Nawawi, S.T., M.T., IPM. dan Bapak Hery Teguh Setiawan, S.T., M.Eng. yang telah memberikan arahan, bimbingan, dan masukan yang sangat berharga dalam penyusunan skripsi ini. Tanpa bantuan dan dorongan beliau, penulisan skripsi ini tidak akan selesai dengan lancar.
6. Ir. Bagus Fatkhurrozi, S.T., M.T., IPM., selaku dosen penguji yang selalu memberikan waktu untuk semua saran dan masukan selama tugas akhir ini.
7. Alifia Revan Prananda, M.Eng., yang selalu meluangkan waktu memberi saran dan masukan pada skripsi penulis.
8. Kakak, kakak ipar dan keponakan, Isti Amalia, Angga Riswanto dan Nabhan Abyasa Anggara yang selalu mendukung penulis.
9. Bapak Ibu Dosen Jurusan Teknik Elektro, Fakultas Teknik, Universitas Tidar yang telah memberi bekal ilmu dari semester awal hingga semester akhir ini.

10. Staf karyawan, petugas perpustakaan Universitas Tidar, BAAK, dan Biro Keuangan yang telah memberikan kemudahan dan kelancaran secara administrasi.
11. Teman-teman Teknik Elektro Universitas Tidar angkatan 2018 yang selalu membantu kelancaran penulisan skripsi penulis.
12. Sahabat terbaik saya Devan dan Ucup yang membantu dan menemani mengerjakan bersama skripsi penulis sehingga dapat terselesaikan dengan memuaskan.
13. Ana Erina Wati, yang juga memberikan semangat dan motivasi serta tempat berbagi cerita dan keluh kesah.
14. Sahabat dari group whatsapp pro player yang selalu berbagi informasi terkini.
15. Teman-teman dari lab 6 dan lab 5 jurusan teknik elektro yang memberikan ide-ide dan selalu menemani penulis mengerjakan skripsi sampai larut malam.
16. Semua pihak yang telah membantu dan mendukung penyusunan skripsi penulis yang tidak bisa penulis sebutkan satu persatu

Terakhir, penulis menyadari bahwa penulisan skripsi ini masih jauh dari sempurna. Oleh karena itu, penulis dengan rendah hati menerima kritik, saran, dan masukan yang membangun untuk penyempurnaan skripsi ini di masa mendatang.

Terimakasih.

Magelang, 2023

Ahmad Robih Rizqi
NPM. 1810501077

MOTTO

“Ayo pergi bersantai!”.

-- Akai --

“Waktu itu bagaikan pedang, jika engkau tidak menggunakannya dengan baik, ia akan memotongmu”.

-- Imam Syafii --

“bersyukur dan jangan sesali hidupmu, bisa jadi kehidupan yang kamu sesali sekarang adalah cita-cita bagi orang lain”.

- - Ahmad Robih Rizqi –



DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
LEMBAR PENGUJI.....	iii
LEMBAR PERNYATAAN.....	iv
KATA PENGANTAR	v
MOTTO	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
INTISARI.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan.....	4
BAB II DASAR TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori.....	7
2.2.1 <i>Internet of Things</i>	7
2.2.2 Sensor Suhu	8
2.2.3 Sensor Kekeruhan	8
2.2.4 Sensor TDS (<i>Total Dissolved Solids</i>)	9
2.2.5 <i>Database</i>	10
2.2.6 MYSQL	11
2.2.7 MongoDB	12
2.2.8 Modul ESP32.....	12

2.2.9 Dynatrace Oneagent.....	13
2.2.10 InfluxDB	14
2.2.11 Apache Jmeter.....	15
2.2.12 Runtime.....	16
2.2.13 Throughput.....	16
2.2.14 CPU Usage	17
2.2.15 Memory Usage	17
2.2.16 Disk I/O.....	17
BAB III METODOLOGI PENELITIAN.....	19
3.1 Alat dan Bahan	19
3.2 Metodologi Penelitian	20
3.2.1 Studi Literatur	21
3.2.2 Parameter pengujian	21
3.2.3 Pembangunan Lingkungan Uji	21
3.2.4 Pengujian dan Pengambilan Data	24
BAB IV HASIL DAN PEMBAHASAN	25
4.1 Pengujian <i>Error</i> sensor.....	25
4.2 Pengujian Penyimpanan Data Sensor.....	26
4.2.1 Hasil pengujian penyimpanan <i>database</i> MySQL	27
4.2.2 Hasil pengujian penyimpanan <i>database</i> MongoDB	27
4.2.3 Hasil pengujian penyimpanan <i>database</i> InfluxDB	28
4.3 Pengujian Penyimpanan Data Gambar	28
4.3.1 Hasil pengujian penyimpanan <i>database</i> MySQL	29
4.3.2 Hasil pengujian penyimpanan <i>database</i> MongoDB	29
4.3.3 Hasil pengujian penyimpanan <i>database</i> InfluxDB.....	30
4.4 Pengujian Performa <i>Database</i> Mengolah Data.....	30
4.4.1 Hasil pengujian performa <i>database insert</i> data sensor	30
4.4.2 Hasil pengujian performa <i>database insert</i> data gambar	32
4.5 Analisis Hasil Pengujian	34
4.5.1 Runtime MySQL, MongoDB dan InfluxDB	34
4.5.2 Throughput MySQL, MongoDB dan InfluxDB	36

4.5.3	<i>CPU usage</i> MySQL, MongoDB dan InfluxDB	37
4.5.4	<i>Memory usage</i> MySQL, MongoDB dan InfluxDB	38
4.5.5	<i>Disk I/O</i> MySQL, MongoDB dan InfluxDB	39
BAB V PENUTUP.....		41
5.1	Kesimpulan.....	41
5.2	Saran	42
DAFTAR PUSTAKA		43



DAFTAR GAMBAR

Gambar 2.1 Sensor Suhu DS18B20.....	8
Gambar 2.2 Sensor Kekeruhan SEN0189.....	9
Gambar 2.3 Sensor TDS	10
Gambar 2.4 Tampilan database MySQL	11
Gambar 2.5 Tampilan database MongoDB	12
Gambar 2.6 Esp32-CAM (Sumber : Datasheet ESP32CAM)	13
Gambar 2.7 Tampilan software Dynatrace Oneagent.....	14
Gambar 2.8 Tampilan InfluxDB.....	15
Gambar 2. 9 Tampilan Apache Jmeter	15
Gambar 3.1 Diagram Alir Penelitian	20
Gambar 3.2 Topologi jaringan.....	21
Gambar 3.3 Diagram blok sistem	22
Gambar 3.4 Flowchart pengujian.....	23
Gambar 4.1 data sensor dalam <i>database</i> MySQL	27
Gambar 4.3 data sensor dalam <i>database</i> InfluxDB	28
Gambar 4.4 data gambar dalam <i>database</i> MySQL.....	29
Gambar 4.5 data gambar dalam <i>database</i> MongoDB	29
Gambar 4.6 data gambar dalam <i>database</i> InfluxDB	30
Gambar 4.7 Grafik performa <i>database</i> dalam mengolah data sensor.....	32
Gambar 4.8 Grafik performa <i>database</i> dalam mengolah data gambar.....	34

DAFTAR TABEL

Tabel 2.1 Spesifikasi sensor kekeruhan	9
Tabel 2.2 Spesifikasi dari sensor TDS V1.0	10
Tabel 4.1 Hasil pengujian data sensor MySQL	27
Tabel 4.2 Hasil pengujian data sensor MongoDB	28
Tabel 4.3 Hasil pengujian data sensor InfluxDB	28
Tabel 4.4 Hasil pengujian data gambar MySQL	29
Tabel 4.5 Hasil pengujian data gambar MongoDB	30
Tabel 4.6 Hasil pengujian data gambar InfluxDB	30
Tabel 4.7 Hasil pengujian runtime data sensor	31
Tabel 4.8 Hasil pengujian runtime data gambar.	32
Tabel 4.9 Hasil pengujian throughput data sensor.....	32
Tabel 4.10 Hasil pengujian throughput data gambar.	35
Tabel 4. 11 Hasil pengujian <i>CPU usage</i> data sensor.....	36
Tabel 4.12 Hasil pengujian <i>CPU usage</i> data gambar.	37
Tabel 4.13 Hasil pengujian <i>Memory usage</i> data sensor.....	39
Tabel 4.15 Hasil pengujian <i>Disk I/O</i> data sensor.....	42

INTISARI

Sensor IoT menghasilkan volume yang sangat besar dan sangat bervariasi yang perlu diproses dalam basis data yang sesuai. Penelitian ini berfokus pada membangun media penyimpanan, menguji performa dan membandingkan performa database MySQL, MongoDB, dan InfluxDB sebagai media penyimpanan data pada IoT. Berdasarkan rumusan masalah maka tujuan dari penelitian ini adalah untuk membangun data store IoT menggunakan database MySQL, MongoDB, dan InfluxDB, kemudian menguji dan membandingkan performa database MySQL, MongoDB, dan InfluxDB sebagai data store IoT. Parameter uji yang digunakan dalam penelitian ini meliputi Runtime, Throughput, CPU usage, Memory usage dan Disk I/O. Data pengujian diambil dari data sensor IoT monitoring kualitas air yaitu sensor pH sen0161, sensor suhu DS18B20, sensor kekeruhan sen0189 dan modul kamera. Ditinjau dari data parameter pengujian, untuk mengolah data gambar MySQL lebih unggul dengan rata-rata runtime 1,656 s, untuk InfluxDB 2,766 s dan MongoDB 4,001 s. Parameter throughput paling unggul MySQL 2,66 ops/s, InfluxDB sebesar 0,58 ops/s dan MongoDB 0,65 ops/s. Parameter CPU Usage paling unggul InfluxDB dengan 27,1%, untuk MySQL 38,7% dan MongoDB 41,1%. Parameter Memory Usage paling unggul MongoDB 84,75 MB, untuk MySQL 158,85 MB dan InfluxDB 525,73 MB. Parameter Disk I/O paling unggul MySQL dengan 133,61 KB/s, untuk InfluxDB 92,36 KB/s dan MongoDB 2,6 KB/s. Hasil parameter menunjukkan MySQL menjadi solusi untuk media penyimpanan data dengan jenis data gambar IoT yang telah dibangun, sedangkan untuk jenis data sensor, InfluxDB menjadi solusi untuk media penyimpanan data sensor IoT yang telah dibangun, dikarenakan InfluxDB merupakan database yang dirancang khusus untuk menangani data *time-series* seperti data sensor IoT.

Kata kunci : IoT, MySQL, MongoDB, InfluxDB

ABSTRACT

The IoT sensors generate a large and highly varied volume of data that needs to be processed in a suitable database. This research focuses on building a storage solution, testing the performance, and comparing the performance of MySQL, MongoDB, and InfluxDB databases as storage media for IoT data. Based on the research problem, the objective of this study is to build an IoT data store using MySQL, MongoDB, and InfluxDB databases, and then test and compare the performance of these databases as IoT data stores. The test parameters used in this research include Runtime, Throughput, CPU usage, Memory usage, and Disk I/O. The testing data is collected from IoT sensors monitoring water quality, specifically the pH sensor sen0161, the temperature sensor DS18B20, the turbidity sensor sen0189, and the camera module. Based on the test parameter data, when processing image data, MySQL performs the best with an average runtime of 1.656 seconds, followed by InfluxDB with 2.766 seconds, and MongoDB with 4.001 seconds. The highest throughput parameter is achieved by MySQL with 2.66 operations per second (ops/s), followed by InfluxDB with 0.58 ops/s, and MongoDB with 0.65 ops/s. In terms of CPU usage, InfluxDB has the lowest value at 27.1%, followed by MySQL at 38.7%, and MongoDB at 41.1%. For memory usage, MongoDB performs the best with 84.75 MB, followed by MySQL with 158.85 MB, and InfluxDB with 525.73 MB. In terms of Disk I/O, MySQL has the highest performance with 133.61 KB/s, followed by InfluxDB with 92.36 KB/s, and MongoDB with 2.6 KB/s. The results of these parameters indicate that MySQL is the preferred solution for storing image data in the IoT, while InfluxDB is the recommended choice for storing sensor data in the IoT, as it is a database specifically designed to handle time-series data such as IoT sensor data.

Keyword: IoT, MySQL, MongoDB, InfluxDB

BAB I

PENDAHULUAN

1.1 Latar Belakang

Konsep *Internet of Things* (IoT) mengacu pada tiga elemen utama: objek fisik yang dilengkapi dengan modul IoT, perangkat yang terhubung ke Internet seperti modem dan router, dan pusat data cloud di mana aplikasi dan penyimpanan data disimpan. Ketika membahas data, kita tidak dapat memisahkan data dari media penyimpanan data, yang dalam hal ini disebut dengan *database* (Pramukantoro. dkk., 2017).

Ada beberapa tantangan dalam membangun media penyimpanan data IoT, antara lain volume data yang besar dan bentuk serta format data yang beragam. IoT menggunakan sejumlah besar sensor dalam kehidupan sehari-hari. Sensor IoT menghasilkan volume yang sangat besar dan sangat bervariasi yang perlu diproses dalam basis data yang sesuai. IoT menghadirkan tantangan bagi basis data dalam menilai cara menyimpan dan memanipulasi data yang sangat bervariasi dalam jumlah besar, (Qodarbaskoro dkk., 2021). Menurut db-engines.com, diperbarui pada tahun 2023, basis data yang paling umum digunakan dalam aplikasi IoT adalah MySQL, MongoDB dan InfluxDB. Penelitian ini bertujuan untuk memberikan evaluasi komparatif yang komprehensif terhadap tiga database yang umum digunakan pada aplikasi IoT, yaitu MySQL, MongoDB, dan InfluxDB.

Ada beberapa jenis database yang dapat Anda gunakan untuk menyimpan dan memproses data sensor IoT. Jenis database yang berbeda memiliki mekanisme implementasi, karakteristik penyimpanan, serta metode konfigurasi dan pengoptimalan yang berbeda. Hal-hal ini harus dipertimbangkan ketika memilih database untuk digunakan sebagai media penyimpanan data sensor IoT. Media penyimpanan data sangat penting untuk memaksimalkan kinerja dengan menangani beragam data dalam IoT.

Hergika (2021) mengatakan bahwa *database* berfungsi sebagai wadah untuk menyimpan data dari berbagai perangkat dan sensor yang terhubung ke jaringan. Penggunaan *database* dalam IoT sangat penting untuk mengelola volume data yang

besar, memastikan keamanan data, dan memberikan pemrosesan *real-time* yang diperlukan untuk mendukung pengambilan keputusan yang tepat waktu.

Salah satu platform *database* yang dapat digunakan untuk IoT adalah MySQL. MySQL adalah sistem manajemen basis data relasional open source yang sering digunakan dalam berbagai aplikasi yang membutuhkan manajemen data yang andal. MySQL memiliki keunggulan dalam hal skalabilitas, kehandalan, kemampuan untuk mengintegrasikan data dari berbagai sumber, dan fitur-fitur keamanan yang mumpuni, Parlaungan S. and Sudrajat (2020).

MySQL merupakan sistem manajemen *database* relasional yang telah lama digunakan dan memiliki kestabilan dan kinerja yang baik. MongoDB merupakan *database* NoSQL yang memungkinkan penyimpanan data dalam format dokumen JSON yang fleksibel dan dapat diakses secara mudah, sedangkan InfluxDB adalah *database* NoSQL yang dirancang khusus untuk data time series, sehingga sangat cocok untuk aplikasi IoT yang menghasilkan data dalam waktu nyata.

MongoDB adalah sistem manajemen basis data non-relasional *open source* yang terkenal dengan kemampuannya untuk mengelola data semi-terstruktur yang dihasilkan oleh perangkat IoT. MongoDB juga memiliki kemampuan skalabilitas yang tinggi dan mudah dikembangkan (Kusumawardhana dkk., 2018). MongoDB memiliki waktu eksekusi yang sangat cepat, untuk menyimpan 50 sampai 1500 data diperlukan waktu 0.2 detik.

InfluxDB adalah sistem manajemen basis data deret waktu open source yang dirancang khusus untuk mengelola data deret waktu seperti data sensor dan log peristiwa. InfluxDB juga memiliki kemampuan untuk mengintegrasikan data dari berbagai sumber dan menangani data dengan skala yang sangat besar (Andriyani, 2020). InfluxDB lebih fleksibel untuk melakukan input maupun output data dalam bentuk visualisasi.

Parameter pengujian yang digunakan dalam penelitian ini adalah sebagai berikut : runtime, throughput, *CPU usage*, *Memory usage*, dan disk I/O. Selain itu, akan diuji juga kemampuan *database* MySQL, MongoDB dan InfluxDB untuk memproses *read* dan *write* berbagai macam data. Penelitian ini bertujuan untuk

mengevaluasi kinerja database MySQL, MongoDB dan InfluxDB, sehingga dapat menjadi acuan bagi mereka yang ingin membangun sistem dengan kebutuhan penyimpanan data, serta dapat menjadi pembandingan dan referensi untuk penelitian di masa selanjutnya.

1.2 Rumusan Masalah

Dari uraian latar belakang yang telah dijabarkan, dapat disimpulkan bahwa Internet of Things (IoT) membutuhkan suatu database sebagai wadah untuk menampung dan memproses data yang berasal dari sensor. Sekarang ini, terdapat beragam jenis database yang dapat dimanfaatkan untuk menyimpan data dari sensor, seperti MySQL, MongoDB dan InfluxDB. Masing-masing jenis *database* memiliki mekanismenya masing-masing. Berdasarkan persoalan yang telah dijelaskan maka dilakukan pengujian kinerja dari masing-masing *database* untuk mengetahui *database* yang sesuai dengan fungsi alat yang akan dibuat.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah maka tujuan dari penelitian ini adalah untuk membangun data store IoT menggunakan database MySQL, MongoDB, dan InfluxDB, kemudian menguji dan membandingkan performa database MySQL, MongoDB, dan InfluxDB sebagai data store IoT.

1.4 Manfaat Penelitian

Penelitian ini memiliki beberapa manfaat yang diharapkan, antara lain :

- a. Penelitian ini diharapkan dapat memberikan informasi mengenai performa database yang telah diteliti, sehingga dapat menjadi referensi bagi mereka yang ingin membangun sistem dengan kebutuhan penyimpanan data yang beragam, dan
- b. Penelitian ini diharapkan dapat membangun data *store* IoT menggunakan database MySQL, MongoDB, dan InfluxDB untuk dapat dijadikan referensi pembuatan media penyimpanan data IoT.

1.5 Batasan Masalah

Ada beberapa batasan masalah yang diterapkan pada penelitian ini, yaitu :

- a. Parameter uji yang digunakan dalam penelitian ini meliputi Runtime, Throughput, CPU usage, Memory usage dan Disk I/O, dan

- b. Data pengujian diambil dari data sensor IoT monitoring kualitas air yaitu sensor TDS, sensor suhu DS18B20, sensor kekeruhan sen0189 dan modul kamera.

1.6 Sistematika Penulisan

Penulisan pada penelitian ini terstruktur dalam 5 bab, yang terdiri dari :

BAB I PENDAHULUAN

Bab pertama dari penelitian ini membahas tentang latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, hipotesis, serta sistematika penulisan.

BAB II DASAR TEORI

Bab kedua pada penelitian ini berisi tentang landasan teori yang digunakan oleh penulis dalam menyusun tugas akhir ini, dengan mengacu pada berbagai referensi yang terkait dengan penelitian ini.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tahapan-tahapan yang akan dilakukan untuk menyelesaikan permasalahan yang ada sesuai dengan judul penelitian, serta metode yang akan digunakan yang disajikan dalam bentuk diagram alir atau format lainnya.

BAB IV HASIL DAN ANALISA

Bab ini menguraikan hasil dari penelitian yang dilakukan sesuai alur pada bab metode penelitian. Hasil tersebut dianalisis untuk kemudian ditarik kesimpulan.

BAB V KESIMPULAN

Bab ini menguraikan kesimpulan yang diambil dari hasil dan analisis penelitian yang telah dilakukan.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Adam Kuku Kurniawan (2018) membandingkan dua database sebagai media penyimpanan data sensor IoT. Database yang diteliti adalah database Cassandra dan database MongoDB. Kedua database tersebut merupakan database berjenis NoSQL atau database non-relasional. Penelitian ini menggunakan beberapa parameter pengujian, seperti runtime, throughput, *memory usage*, *CPU usage*, dan *disk I/O*. Data yang digunakan pada operasi *insert* meliputi data string dan data gambar dengan jumlah yang bervariasi, yaitu 10.000, 30.000, 50.000, 70.000, 100.000, 120.000, dan 150.000 untuk data string, serta 1000, 3000, 5000, 7000, dan 10.000 untuk data gambar. Hasil penelitian tersebut menyimpulkan bahwa NoSQL Cassandra adalah pilihan yang baik untuk penyimpanan data jika ukuran data yang dipertukarkan besar, tetapi NoSQL MongoDB adalah pilihan yang baik untuk penyimpanan data jika ukuran data yang dipertukarkan kecil.

Aziiz, Ichsan and Arwani (2017) dalam penelitiannya menggunakan data sensor LM35 untuk membandingkan kinerja antara database MySQL dan MongoDB. Parameter uji yang digunakan adalah perbandingan waktu *insert* data sensor LM35 ke database MySQL dan MongoDB. Pengujian terhadap sistem dilakukan dengan menguji pengiriman ke 1 tabel atau koleksi, 2 tabel atau koleksi, dan 3 tabel atau koleksi ke masing - masing basis data selama 10 kali pengiriman sehingga didapatkan waktu eksekusi *query insert* sebagai parameter pembanding basis data. Hasil dari penelitian ini menunjukkan bahwa database MongoDB secara signifikan lebih baik daripada database MySQL dalam hal waktu eksekusi untuk kueri *insert* data sensor LM35.

Wicaksono (2018) meneliti kinerja Hbase dan MongoDB untuk perbandingan kinerja basis data non-relasional. Penelitian ini menggunakan beberapa parameter pengujian, antara lain runtime, throughput, memory usage, CPU usage, dan disk I/O. Data yang digunakan dalam melakukan operasi *insert* adalah data teks dan gambar, dan untuk pengujian teks, variasi data yang digunakan adalah 10.000,

30.000, 50.000, 70.000, dan 100.000. Sedangkan pengujian gambar memakai lima variasi data yaitu 1000, 3000, 5000, 7000, 10.000. Pada pengujian kinerja, MongoDB terbukti unggul dalam parameter runtime dengan hasil 33 detik, throughput 1559 ops/s, CPU usage 48%, memory usage 67%, dan disk I/O sebesar 4354kb. Namun, saat dilakukan pengujian pada data gambar, MongoDB masih unggul pada parameter runtime dengan hasil 64 detik dan throughput 90 ops/s, serta CPU usage 46% dan memory usage 84%. Sedangkan pada parameter disk I/O, Hbase terbukti unggul dengan hasil 109.462kb.

Putra dan , Oktania Purwaningrum (2022) meneliti tentang perbandingan performa respon waktu *query* MySQL, postgresql, dan MongoDB. Parameter yang digunakan dalam penelitian ini adalah waktu yang diperlukan setiap DBMS dalam mengeksekusi *query*. Penelitian ini melakukan eksekusi pada berbagai skenario *query*, yaitu *INSERT (create)*, *SELECT (read)*, *UPDATE (update)*, *DELETE (delete)*, *SUM*, dan *COUNT*. Hasil dari skenario pengujian, *database* PostgreSQL paling unggul dibandingkan MySQL dan MongoDB berdasarkan total *response time*. Penelitian ini membuktikan bahwa stigma yang mengatakan bahwa performa dari NoSQL lebih cepat dibandingkan dengan DBMS tidak selamanya benar.

Qodarbaskoro dan Utam (2021) membahas mengenai penggunaan basis data relasional dan non-relasional pada aplikasi IoT (Internet of Things). Basis data relasional yang menggunakan bahasa SQL disebutkan dalam banyak penelitian memiliki keunggulan pada stabilitas dan keamanan data. Sedangkan basis data non-relasional atau NoSQL memiliki keunggulan pada kecepatan, skalabilitas, dan fleksibilitas penyimpanan, terutama untuk data yang bersifat heterogen. Pemilihan basis data untuk sistem IoT yang akan dibangun sangat ditentukan oleh kriteria dan spesifikasi sistem tersebut. Basis data relasional (SQL) memiliki keunggulan dalam menjaga konsistensi data, keamanan, dan kinerja yang baik pada volume data yang relatif kecil. Sementara itu, basis data non-relasional (NoSQL) menawarkan fleksibilitas penyimpanan data yang cocok untuk data yang heterogen, skalabilitas, dan kinerja yang lebih cepat pada volume data yang lebih besar.

Eyada dan Genidy (2020) membandingkan kinerja *database* IoT menggunakan *database* MongoDB dan MySQL. Skenario pengujian dilakukan

dengan cara *insert* 4000 data sebanyak dua belas kali. Parameter yang diukur adalah latency dan ukuran *database*. Dari pengujian menghasilkan bahwa MongoDB memiliki keunggulan latency yang lebih kecil dibandingkan MySQL, sedangkan MySQL memiliki keunggulan dari segi penyimpanan yang lebih kecil dibandingkan MongoDB.

Asiminidis (2018) membandingkan dua basis data: relasional dan non-relasional. Basis data relasional yang diteliti adalah MySQL dan PostgreSQL, dan basis data non-relasional adalah MongoDB. Parameter yang diuji adalah waktu *insert queries*. Hasil dari penelitian ini adalah saat data uji sedikit PostgreSQL lebih unggul dibandingkan dengan MySQL dan MongoDB. MongoDB lebih unggul dari MySQL dan PostgreSQL ketiga data uji lebih banyak. MySQL lebih unggul dari PostgreSQL ketika data uji lebih dari 20000 data uji, namun masih belum dapat mengungguli MongoDB.

Makris, dkk (2021) melakukan perbandingan *database* MongoDB dengan PostgreSQL dengan aspek kinerja sebagai tolok ukur. Dari semua hasil skenario pengujian, PostgreSQL lebih unggul dibandingkan MongoDB. Selanjutnya, waktu repon rata-rata berkurang secara signifikan ketika kedua *database* menggunakan indeks, terutama dalam kasus MongoDB.

Patil, dkk (2017) melakukan analisa dari kinerja MongoDB dan MySQL berdasarkan *insert* dan pengambilan data menggunakan aplikasi web untuk menguji kecepatan *response time*. Data penelitian yang digunakan adalah jumlah *request* atau permintaan untuk mngambil data yang disimpan didalam *database*. Hasil menunjukkan *database* MongoDB memiliki response time lebih kecil dibandingna MySQL disemua pengujian dengan jumlah *request* atau permintaan yang berbeda-beda.

2.2 Dasar Teori

Dalam penulisan ini, beberapa teori telah dipaparkan berdasarkan informasi dan kajian pustaka yang telah dilakukan, antara lain :

2.2.1 Internet of Things

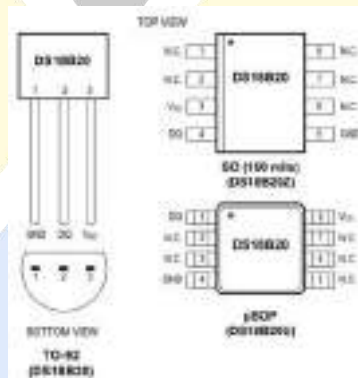
IoT atau Internet of Things adalah konsep dimana semua objek fisik di dunia dapat saling berhubungan dan berkomunikasi sebagai bagian dari sistem

terintegrasi menggunakan internet sebagai media penghubung. Pada dasarnya perangkat IoT terdiri dari sensor sebagai media pengumpul data, sambungan internet sebagai media komunikasi, dan *server* sebagai pengumpul informasi yang diterima sensor untuk dianalisa.

2.2.2 Sensor Suhu

Sensor suhu berfungsi untuk mengonversi perubahan suhu menjadi sinyal listrik dalam bentuk tegangan. Sensor suhu DS18B20 dipilih sebagai sensor dalam monitoring suhu ini dikarenakan selain stabil, aman, handal, serta *user friendly* dalam hal pemasangan, juga dapat digunakan untuk melakukan pengukuran dengan *range* -55 to +125°C tanpa kalibrasi, dengan menggunakan 3 kaki yang berguna untuk GND, VCC dan 1- wire data. Sensor suhu DS18B20 menyediakan 9-12 bit untuk pengukuran suhu dalam celcius (°C). Oleh karena itu sensor suhu DS18B20 sangat cocok digunakan dalam sistem ini (Semiconductor, 2008).

Sensor suhu yang digunakan dalam penelitian ini adalah sebuah komponen elektronik yang diproduksi oleh Maxim Integrated Products. Sensor suhu tersebut dapat dikenali dari tampilan fisik dan susunan pin-nya yang terdapat pada Gambar 2.2. Komponen ini dinamakan sebagai sensor suhu DS18B20.



Gambar 2.1 Sensor Suhu DS18B20

Sumber: (DallasDS18B20 Temperature Sensors Semiconductor Datasheet, 2002).

2.2.3 Sensor Kekeruhan

Sistem sensor ini berfungsi untuk mendeteksi tingkat kekeruhan air dengan menggunakan detector dan sumber cahaya. Fotodioda TSL 250 digunakan sebagai detector yang sensitif terhadap perubahan intensitas cahaya yang diterimanya.

Sumber cahaya yang digunakan adalah dioda laser yang akan menembakkan cahaya ke dalam air. Jika air tersebut memiliki banyak partikel (keruh), sebagian cahaya akan diteruskan dan sebagian akan dihamburkan. Intensitas cahaya yang diterima oleh fotodioda TSL 250 akan mengukur intensitas cahaya yang dihamburkan oleh partikel dalam air. Kemudian, intensitas cahaya tersebut akan dikonversi menjadi sinyal tegangan, yang menunjukkan nilai tegangan yang sebanding dengan tingkat kekeruhan air (Sasmoko, Rasminto dan Rahmadani, 2019).



Gambar 2.2 Sensor Kekeruhan SEN0189
Sumber: Datasheet Sen0189 DFROBOT

Tabel 2.1 Spesifikasi sensor kekeruhan

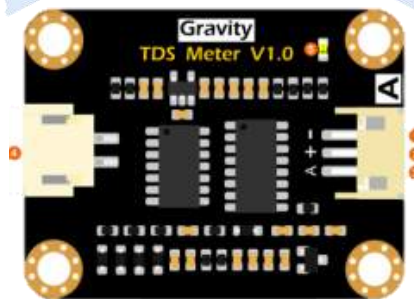
No	Nama	Keterangan
1	Operating Voltage	5V DC
2	Operating Current:	40mA (MAX)
3	Response Time	<500ms
4	Insulation Resistance	100M (Min)
5	Operating Temperature	5°C~90°C
6	Storage Temperature	-10°C~90°C
7	Weight	30g
8	Adapter Dimensions	38mm*28mm*10mm/1.5inches*1.1inches*0.4inches

2.2.4 Sensor TDS (*Total Dissolved Solids*)

Total Dissolved Solids (TDS) mengindikasikan jumlah zat terlarut dalam satu liter air, diukur dalam miligram. Semakin tinggi nilai TDS, semakin banyak zat terlarut yang terdapat dalam air, yang menunjukkan semakin rendahnya

kebersihan air tersebut. Oleh karena itu, nilai TDS dapat menjadi referensi untuk mengevaluasi tingkat kebersihan air.

Sensor TDS bekerja dengan prinsip dua elektroda yang diberikan arus searah, dan perubahan nilai konduktivitas listrik dibaca sebagai nilai tegangan. Elektroda sensor konduktivitas dimasukkan ke dalam sampel air untuk mendapatkan nilai TDS. Probe TDS tahan air sehingga dapat direndam dalam air untuk waktu yang lama. Sensor gravity TDS memiliki bentuk fisik seperti yang terlihat pada Gambar 2.4 dan spesifikasinya tertera pada Tabel 2.4. (Cahyani, Harmadi, & Wildian, 2016).



Gambar 2.3 Sensor TDS

Sumber: Datasheet SN0244 DFRobot

Tabel 2.2 Spesifikasi dari sensor TDS V1.0

No.	Nama	Keterangan
1	Tegangan input	3.3 ~ 5.5V
2	Tegangan output	0 ~ 2.3V
3	Aktif pada arus	3 ~ 6mA
4	Jangkauan pengukuran <i>TDS</i>	0 ~ 1000ppm
5	Akurasi pengukuran <i>TDS</i>	± 10% F.S. (25 °C)
6	Ukuran modul sensor	42 * 32mm

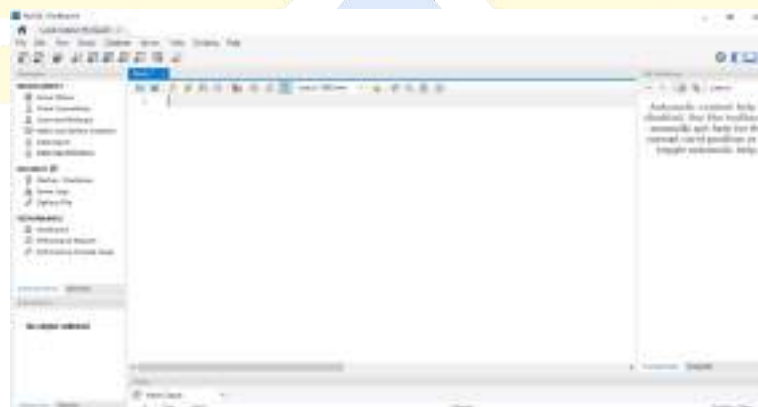
2.2.5 Database

Database adalah sekumpulan data yang merepresentasikan objek-objek dunia nyata dan digunakan untuk tujuan tertentu oleh satu atau lebih pengguna. Data sendiri merupakan fakta-fakta yang diketahui, direkam dan memiliki makna tersirat. Untuk mengelola dan memelihara database secara terkomputerisasi, DBMS

dapat digunakan. DBMS merupakan perangkat lunak yang digunakan untuk membuat dan mengelola *database*. Sebagai contoh, kita dapat menyimpan nama, nomor telepon, dan alamat orang yang dikenal dalam sebuah buku alamat berindeks atau menggunakan perangkat lunak seperti Microsoft Access atau Excel untuk menyimpan data tersebut pada *hard drive* komputer. Dengan demikian, koleksi data dengan makna tersirat ini dapat disebut sebagai *database*, sedangkan perangkat lunak seperti Microsoft Access atau Excel disebut sebagai DBMS (Elmasri dan Navathe, 2011).

2.2.6 MYSQL

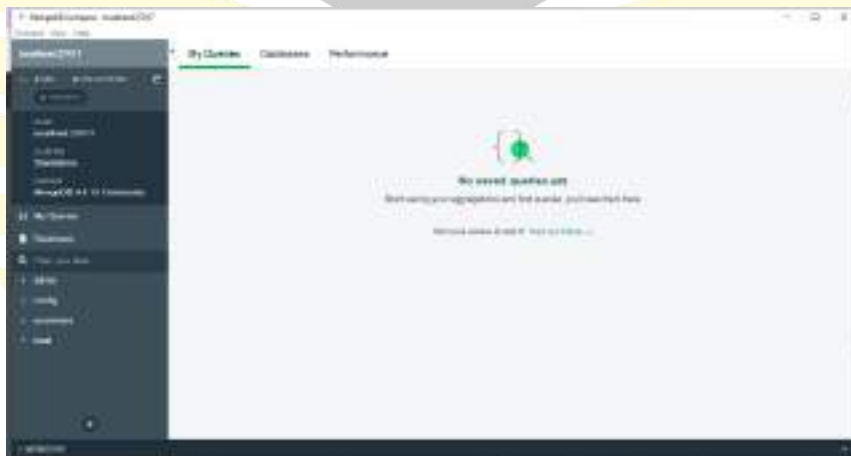
Menurut Bunafit Nugroho (2004:1), MySQL adalah sebuah program *server database* yang dapat dengan cepat menerima dan mengirim data, dapat diakses oleh banyak pengguna secara bersamaan, dan menggunakan standar SQL (*Structure Query Language*). Dengan menggunakan MySQL *server*, pengguna dapat membatasi akses berdasarkan hak pengguna yang diberikan. MySQL menggunakan bahasa pemrograman standar untuk database yaitu SQL (*Structure Query Language*). MySQL pertama kali dipublikasikan pada tahun 1996, tetapi sebenarnya telah dikembangkan sejak tahun 1979. MySQL telah menerima penghargaan Linux Journal Reader's Choice Award selama tiga tahun berturut-turut. Sekarang MySQL tersedia di bawah lisensi *opensource*, namun juga tersedia lisensi untuk penggunaan komersial.



Gambar 2.4 Tampilan database MySQL
(Sumber : Andhikha, 2018)

2.2.7 MongoDB

MongoDB adalah sistem basis data *opensource* berkinerja tinggi yang menggunakan konsep manajemen berorientasi dokumen dan dibangun menggunakan bahasa pemrograman C++. Basis data berorientasi dokumen dirancang untuk menyimpan, mengambil, dan mengelola data dalam format berorientasi dokumen. MongoDB termasuk dalam kategori database NoSQL, yang tidak memerlukan penggunaan perintah-perintah query structured query language (SQL) untuk melakukan manipulasi data. MongoDB adalah database non-relasional, sehingga proses manipulasi data dilakukan dengan cepat dibandingkan dengan sistem database relasional (RDBMS). MongoDB tidak memiliki struktur tabel yang terorganisir seperti database relasional, namun proses pencarian dapat dilakukan lebih cepat, tidak perlu mendesain struktur tabel, dan memiliki kecepatan proses CRUD yang cepat. Oleh karena itu, MongoDB sering digunakan pada website dengan kompleksitas data yang cukup besar. MongoDB tidak memerlukan hubungan antara tabel dan disimpan dalam koleksi, setiap koleksi terdiri dari dokumen yang memiliki field.



Gambar 2.5 Tampilan database MongoDB
(Sumber : Suendri, 2020)

2.2.8 Modul ESP32

ESP32-CAM adalah papan pengembangan WiFi atau Bluetooth dengan mikrokontroler ESP32 dan kamera. ESP32-CAM tidak memiliki memori PSRAM, hanya saja memiliki RAM sebesar 512 kB serta memiliki modul kamera OV2640.

OV2640 merupakan *chip* kamera dengan sensor kamera sebesar 2MP. Selain itu, OV2640 mampu beroperasi hingga 15 frame per detik (fps) serta menggunakan teknologi sensor untuk meningkatkan kualitas gambar dengan mengurangi atau menghilangkan sumber cahaya, sehingga menghasilkan warna gambar yang bersih.



Gambar 2.6 Esp32-CAM

(Sumber : Datasheet ESP32CAM)

2.2.9 Dynatrace Oneagent

Dynatrace OneAgent adalah agen yang dipasang pada host atau aplikasi untuk memantau kinerja dan masalah pada infrastruktur dan aplikasi. Berikut adalah beberapa konsep dan teknologi yang menjadi dasar teori Dynatrace OneAgent:

1. *Code-level Visibility*: Dynatrace OneAgent dapat memantau aktivitas pada level kode, seperti *CPU usage*, memori, dan disk, serta jumlah panggilan fungsi. Dengan demikian, Dynatrace dapat mengetahui dengan detail bagaimana aplikasi berinteraksi dengan infrastruktur.
2. *Real User Monitoring*: Dynatrace OneAgent juga dapat memantau pengguna yang mengakses aplikasi secara langsung dari perangkatnya masing-masing (*Real User Monitoring*). Hal ini memungkinkan Dynatrace untuk melacak pengalaman pengguna (*user experience*) dan mengetahui waktu respon, kesalahan, dan performa aplikasi secara langsung dari perspektif pengguna.
3. *Deep Transaction Tracing*: Dynatrace OneAgent menggunakan teknologi *deep transaction tracing* untuk melacak setiap permintaan (*request*) yang dilakukan pada aplikasi dari awal hingga akhir, termasuk interaksi dengan layanan terkait (contoh: *database* atau layanan web lainnya). Hal ini memungkinkan

Dynatrace untuk memberikan informasi yang akurat dan detail terkait performa dan masalah pada aplikasi.

4. *Smartscape Topology*: Dynatrace OneAgent juga berkontribusi pada konsep Smartscape, dengan memperbaharui informasi keterkaitan antara komponen aplikasi dan infrastruktur yang dipantau. Hal ini memudahkan pengguna untuk memahami keterkaitan antara komponen infrastruktur dan aplikasi, sehingga memudahkan pengguna untuk mengidentifikasi masalah yang terjadi.



Gambar 2.7 Tampilan software Dynatrace Oneagent

(Sumber : TOMASZ GAJGER, 2020)

2.2.10 InfluxDB

InfluxDB merupakan sebuah *database* yang bersifat *open-source* dengan berbasis bahasa GO, InfluxDB digunakan untuk mengumpulkan data matrik dari sensor – sensor IoT dan perintah pada InfluxDB memiliki kesamaan dengan SQL. Didalam InfluxDB terdapat Telegraf yang merupakan *database* yang menjadi subscriber dari MQTT dan melakukan *publish* pada InfluxDB. Grafana merupakan perangkat lunak yang berbasis *open-source* yang sering digunakan untuk melakukan monitoring dan analisis data, Grafana mendukung penggunaan dengan banyak data dan dapat memberikan *monitoring* serta analisis data secara *real-time* Putra et al. (2021).

2.2.12 Runtime

Runtime pada *database* adalah waktu yang dibutuhkan untuk menjalankan suatu *query* atau transaksi pada *database*. Semakin cepat runtime, semakin baik performa *database*. Faktor-faktor yang mempengaruhi runtime pada *database* antara lain ukuran *database*, jumlah pengguna, dan jenis *query* yang digunakan. *Performance tuning* berperan dalam meningkatkan runtime maksimum yang dapat dicapai oleh *database*. Tanpa melakukan *tuning* performa, kinerja *database* dan aplikasi yang sering digunakan akan menurun (Fitri Marisa et al., 2022).

MySQL adalah salah satu *database management system* yang dapat membantu meningkatkan runtime pada *database*. MySQL dapat dipasang pada *server* dengan spesifikasi kecil dan mendukung *multi user*. Oleh karena itu, pemantauan dan pengelolaan runtime pada *database* sangat penting untuk menjaga performa dan stabilitas *database*.

2.2.13 Throughput

Throughput pada *database* adalah ukuran seberapa banyak tindakan yang diselesaikan dalam jangka waktu tertentu. Pada *database*, throughput dapat diukur dengan seberapa cepat *database* dapat memproses transaksi. Semakin tinggi throughput, semakin banyak transaksi yang dapat diproses dalam waktu tertentu (Fitri Marisa et al., 2022).

Faktor-faktor yang mempengaruhi throughput pada *database* antara lain ukuran *database*, jumlah pengguna, dan jenis *query* yang digunakan. *Performance tuning* berperan dalam melakukan tolak ukur throughput maksimum yang dicapai oleh *database*. Tanpa melakukan *tuning* performa, kinerja *database* dan aplikasi yang sering digunakan akan menurun.

MySQL adalah salah satu *database management system* yang mendukung *multi user* dan dapat dipasang pada *server* dengan spesifikasi kecil. MySQL dapat membantu meningkatkan throughput pada *database*. Oleh karena itu, pemantauan dan pengelolaan throughput pada *database* sangat penting untuk menjaga performa dan stabilitas *database*.

2.2.14 CPU Usage

CPU usage atau penggunaan *CPU* adalah istilah yang digunakan untuk menggambarkan seberapa besar prosesor bekerja. *CPU* (*Central Processing Unit*) adalah sebuah perangkat keras komputer yang memahami dan melaksanakan perintah dari data perangkat lunak (Hidayah and Saptono, 2018). *CPU usage* atau penggunaan *CPU* adalah istilah yang digunakan untuk menggambarkan seberapa besar prosesor bekerja.

CPU merujuk pada sebuah perangkat keras komputer yang memahami dan melaksanakan perintah dari data perangkat lunak. *CPU usage* yang terlalu tinggi dapat berdampak fatal bagi komputer hingga menyebabkan kerusakan *hardware*. *CPU Profiler* dapat digunakan untuk memeriksa penggunaan *CPU* dan aktivitas *thread* aplikasi secara real time saat berinteraksi dengan aplikasi.

2.2.15 Memory Usage

Memori adalah tempat penyimpanan sementara data dan instruksi yang sedang diproses oleh *CPU*. Memori dibagi menjadi dua jenis, yaitu *RAM* (*Random Access Memory*) dan *ROM* (*Read Only Memory*). Pada *database*, *memory usage* atau penggunaan memori dapat mempengaruhi performa *database*. Jika penggunaan memori terlalu tinggi, dapat menyebabkan *database* menjadi lambat dan tidak responsive (WilliamDAssafMSFT, 2023).

Pemantauan penggunaan memori pada *server database* sangat penting untuk menjaga stabilitas dan kecepatan *database*. Beberapa *tools* yang dapat digunakan untuk memantau penggunaan memori pada *database* antara lain *SQL Server Management Studio* dan *Performance Monitor*. Oleh karena itu, pemantauan dan pengelolaan penggunaan memori pada *database* sangat penting untuk menjaga performa dan stabilitas *database*.

Pada aplikasi Android, pengelolaan memori sangat penting untuk menjaga performa aplikasi. Beberapa mekanisme yang digunakan untuk mengelola memori pada Android antara lain *kernel swap daemon* dan *low-memory killer*.

2.2.16 Disk I/O

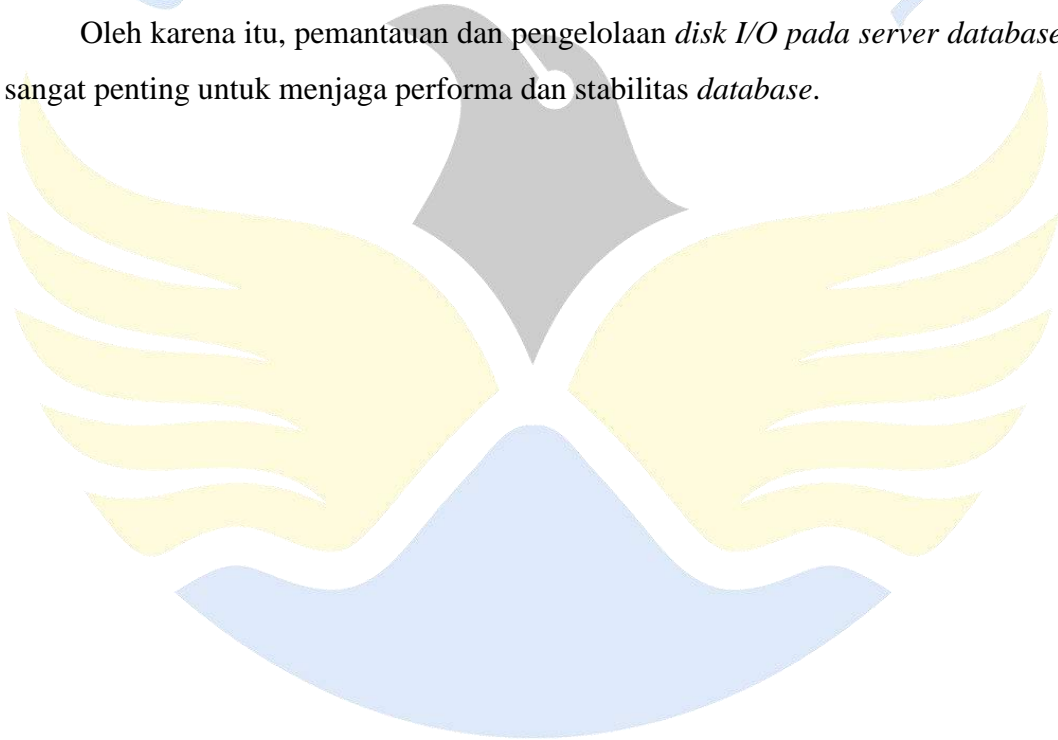
Disk I/O atau *Input/Output Disk* adalah istilah yang digunakan untuk menggambarkan proses membaca dan menulis data pada *disk drive*. *Disk I/O* adalah

salah satu faktor yang mempengaruhi performa *database*. Jika *disk I/O* terlalu lambat, dapat menyebabkan database menjadi *lambat* dan tidak responsive (Ronald Watrianthos and Iwan Purnama, 2018).

Pada database, disk I/O terjadi ketika data harus dibaca atau ditulis dari atau ke *disk drive*. Semakin sering disk I/O terjadi, semakin lambat performa *database*. Beberapa faktor yang dapat mempengaruhi *disk I/O* pada *database* antara lain ukuran *database*, jumlah pengguna, dan jenis *query* yang digunakan.

Untuk mengoptimalkan *disk I/O* pada *database*, dapat dilakukan dengan cara memperbaiki desain *database*, melakukan *indexing*, dan memperbarui *hardware server*. MySQL adalah salah satu *database management system* yang menggunakan *disk I/O* untuk membaca dan menulis data pada *disk drive*. MySQL dapat dipasang pada *server* dengan spesifikasi kecil dan mendukung *multi user*.

Oleh karena itu, pemantauan dan pengelolaan *disk I/O* pada *server database* sangat penting untuk menjaga performa dan stabilitas *database*.



BAB III METODOLOGI PENELITIAN

3.1 Alat dan Bahan

Dalam membuat penelitian ini memerlukan alat dan bahan yang ditunjukkan pada tabel dibawah :

Tabel 3.1 Alat yang digunakan dalam penelitian

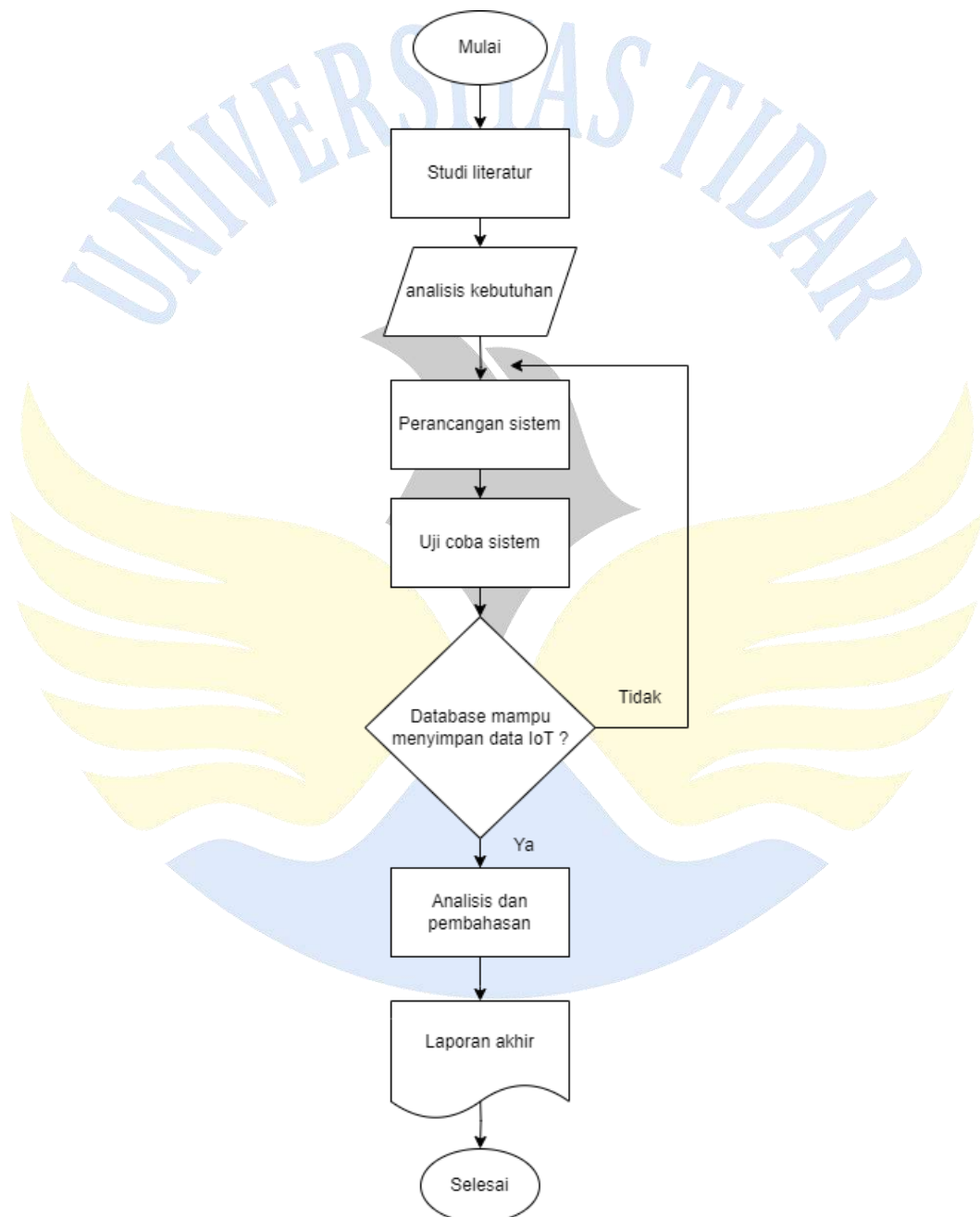
No	Nama	Keterangan
1	ESP32 + modul camera	<i>Mikrokontroller</i>
2	ESP32	<i>Mikrokontroller</i>
3	PC intel(R) core(TM) i3-9100F CPU @3.60GHz	<i>PC Server</i>
4	PC intel(R) core(TM) i5-8250U CPU @1.60GHz	<i>PC Client</i>
5	Sensor TDS sen0244	Input data sensor TDS
6	Sensor kekeruhan sen0189	Input data sensor kekeruhan
7	Sensor suhu DS18B20	Input data sensor suhu
8	MYSQL	Media penyimpanan data
9	MongoDB	Media penyimpanan data
10	InfluxDB	Media penyimpanan data
11	Arduino IDE	<i>Tools pemrograman mikrokontroller</i>
12	Dynatrace Oneagent	<i>Tools monitoring data CPU, Memory dan Disk I/O</i>
13	Apache Jmeter	<i>Tools monitoring Throughput dan Runtime</i>

Tabel 3.2 Bahan yang dibutuhkan

No	Nama	Jumlah	Keterangan
1	Data sensor TDS sen0244	20000	Data untuk variabel penelitian
2	Data sensor kekeruhan sen0189	20000	Data untuk variabel penelitian
3	sensor suhu DS18B20	20000	Data untuk variabel penelitian
4	Data modul camera	20000	Data untuk variabel penelitian
5	Papan PCB	1 (20 cm x 20 cm)	Meletakkan komponen elektronik
6	Kabel Jumper	-	Menghubungkan tiap komponen

3.2 Metodologi Penelitian

Penelitian ini menggunakan beberapa tahapan meliputi survei pendahuluan, studi pustaka, tahapan pengumpulan dan pengolahan data, melakukan perancangan IoT, perancangan model *database*, perancangan lingkungan uji, pengujian dan pengambilan data, analisis pengujian, kesimpulan, dan penyusunan laporan akhir. Diagram alir penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.2.1 Studi Literatur

Dalam melaksanakan penelitian ini, peneliti melakukan studi literatur untuk mempelajari berbagai teori yang mendukung penelitian ini. Refrensi penunjang dalam penelitian ini berasal dari beberapa jurnal penelitian yang membahas tentang performa MySQL dan MongoDB *database*. Selain itu, peneliti juga mendapatkan refrensi melalui buku-buku elektronik (*ebook*) dan situs website yang menjelaskan beberapa teori yang berkaitan dengan penelitian ini.

3.2.2 Parameter pengujian

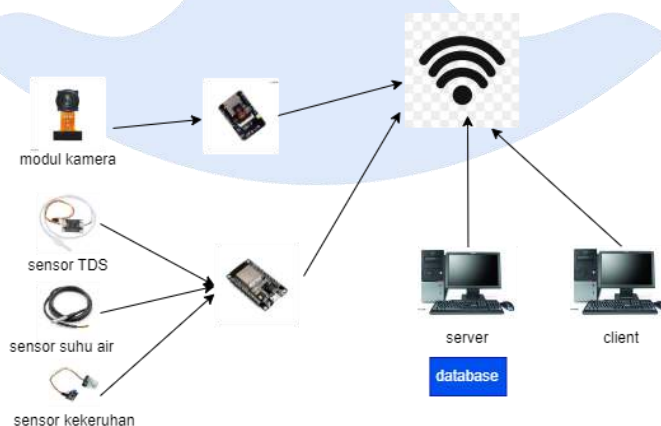
Parameter yang digunakan untuk membandingkan performa *database* adalah Qos dan *Resource Utilization*. Parameter Qos (*Runtime* dan *Throughput*) digunakan untuk mengukur berapa lama waktu yang dibutuhkan *database* untuk menyelesaikan suatu *Query*. Sedangkan parameter *Resource utilization* (*Memory usage*, *CPU usage* dan *Disk I/O*) digunakan untuk mengetahui beban *server* ketika *database* sedang beroperasi. Semakin kecil nilai parameter maka semakin baik *database* saat beroperasi.

3.2.3 Pembangunan Lingkungan Uji

Mengembangkan lingkungan pengujian adalah tahap mempersiapkan berbagai desain yang digunakan dalam proses pengujian. Desain-desain ini dapat dibagi menjadi desain topologi, desain basis data, desain utilitas pengujian, desain skenario pengujian, dan penggunaan data pengujian.

a. Perancangan Topologi jaringan

Topologi jaringan ditunjukkan pada gambar 3.2.



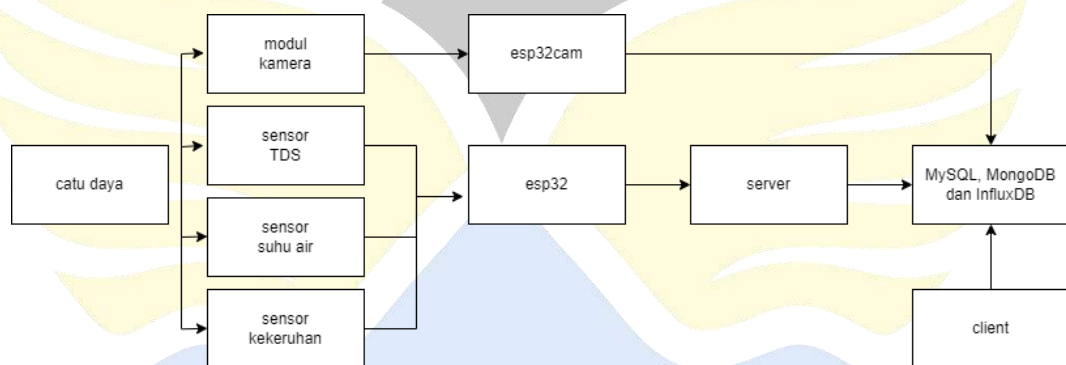
Gambar 3.2 Topologi jaringan

Pada perancangan topologi jaringan, jaringan dibuat dengan kondisi terdapat satu *wireless connection*, dua komputer sebagai *client* dan *server*, satu buah ESP32 sebagai mikrokontroller dan empat buah sensor sebagai sumber data. Topologi pada gambar 3.2 digunakan penulis dalam melakukan implementasi pengujian pada penelitian ini.

Penjelasan topologi :

1. Tiga buah sensor yang terdiri dari sensor suhu DS18B20, sensor kekeruhan Sen0189 dan sensor TDS Sen0244 yang terhubung dengan esp32.
 2. Modul kamera yang terhubung dengan esp32cam.
 3. satu buah esp32 dan satu buah esp32cam yang terhubung dengan *wireless connection*.
 4. 2 buah komputer yang terdiri dari komputer *client* dan *server* yang terhubung dengan *wireless connection*.
- b. Diagram arsitektur pengujian

Diagram blok sistem ditunjukkan pada gambar 3.3.



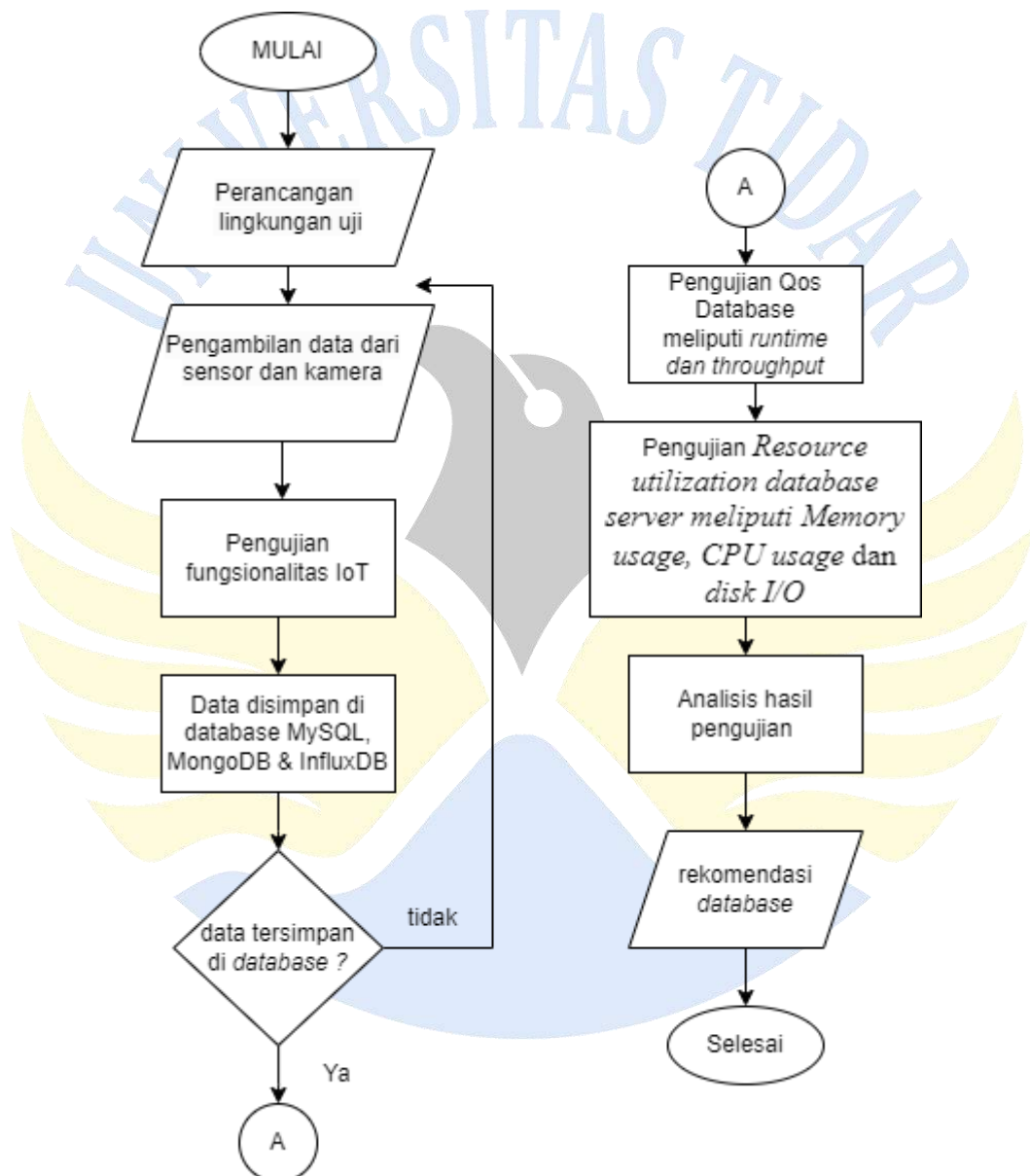
Gambar 3.3 Diagram blok sistem

Pada gambar diagram blok sistem, terlebih dahulu menghubungkan sensor suhu DS18B20, sensor kekruhan sen019 dan sensor TDS sen0244 ke esp32, dan modul kamera dihubungkan ke esp32cam. Kemudian esp32 dan esp32cam disambungkan ke komputer *server*. *Server* akan mengambil sensor suhu DS18B20, sensor kekruhan sen019, sensor TDS sen0244 dan data kamera yang terhubung dan melakukan input atau menyimpan data ke *database* MySQL, MongoDB dan InfluxDB. Pengujian performa dilakukan

sesuai skenario pengujian. Pengujian performa dilakukan dari komputer *server* dan komputer *client* yang terhubung kedalam satu jaringan internet. Hasil data pengujian dapat dianalisa dan digunakan untuk membandingkan performa *database* MySQL dan MongoDB.

c. Diagram Alir Kerja Sistem

Diagram alir kerja sistem ditunjukkan pada Gambar 3.4.

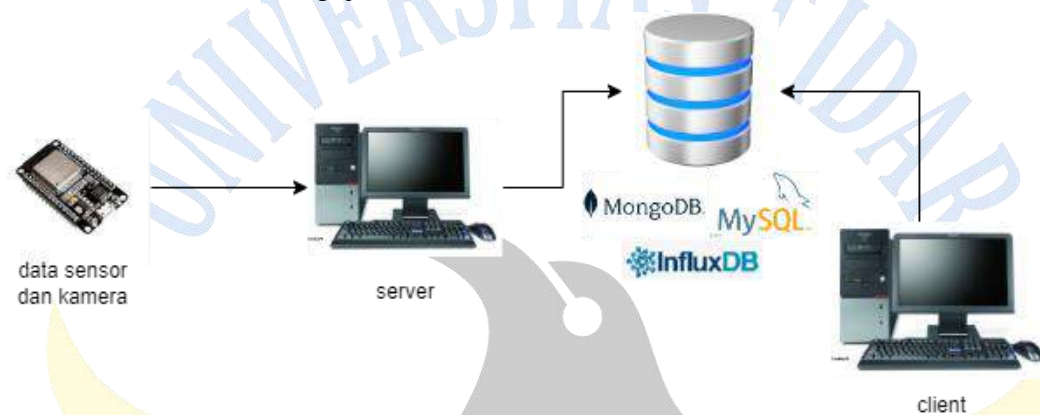


Gambar 3.4 Flowchart pengujian

3.2.4 Pengujian dan Pengambilan Data

Pengujian dilakukan untuk mengumpulkan data informasi mengenai nilai-nilai parameter seperti runtime, throughput, *Memory usage*, *CPU usage*, dan disk I/O ketika melakukan operasi *insert* data dengan jumlah data yang bervariasi. Hasil dari pengujian parameter tersebut digunakan untuk mengukur performa dari MySQL, MongoDB, dan InfluxDB sebagai media penyimpanan data perangkat IoT.

3.2.5 Skenario Pengujian



Gambar 3.5 skenario pengujian

Pengujian performa database dilakukan dengan memeriksa parameter-parameter pengujian seperti runtime, throughput, Memory usage, CPU usage, dan disk I/O ketika melakukan operasi *insert* data pada database MySQL dan MongoDB. Data yang digunakan saat melakukan operasi *insert* berupa data sensor dan data gambar, dengan jumlah data yang bervariasi antara 1.000, 3.000, 5.000, 7.000, 10.000, 15.000, 20.000 untuk data sensor dan 1000, 3000, 5000, 7000, 10.000, 15.000, 20.000 untuk data gambar. *Client* memasukkan data dari sensor dalam bentuk data sensor dan data gambar dari jarak jauh, sesuai skenario, melalui antarmuka web atau IoT ke *server* untuk melakukan operasi *insert* data. Selama proses *insert* data, *CPU usage*, *Memory usage*, dan *Disk I/O* diambil dari *server database* dan dimonitor menggunakan aplikasi Dynatrace Oneagent melalui komputer *client*. Untuk parameter runtime dan throughput diukur sebagai waktu yang dibutuhkan database untuk mengeksekusi sebuah *query* dengan menggunakan aplikasi Apache Jmeter.

DAFTAR PUSTAKA

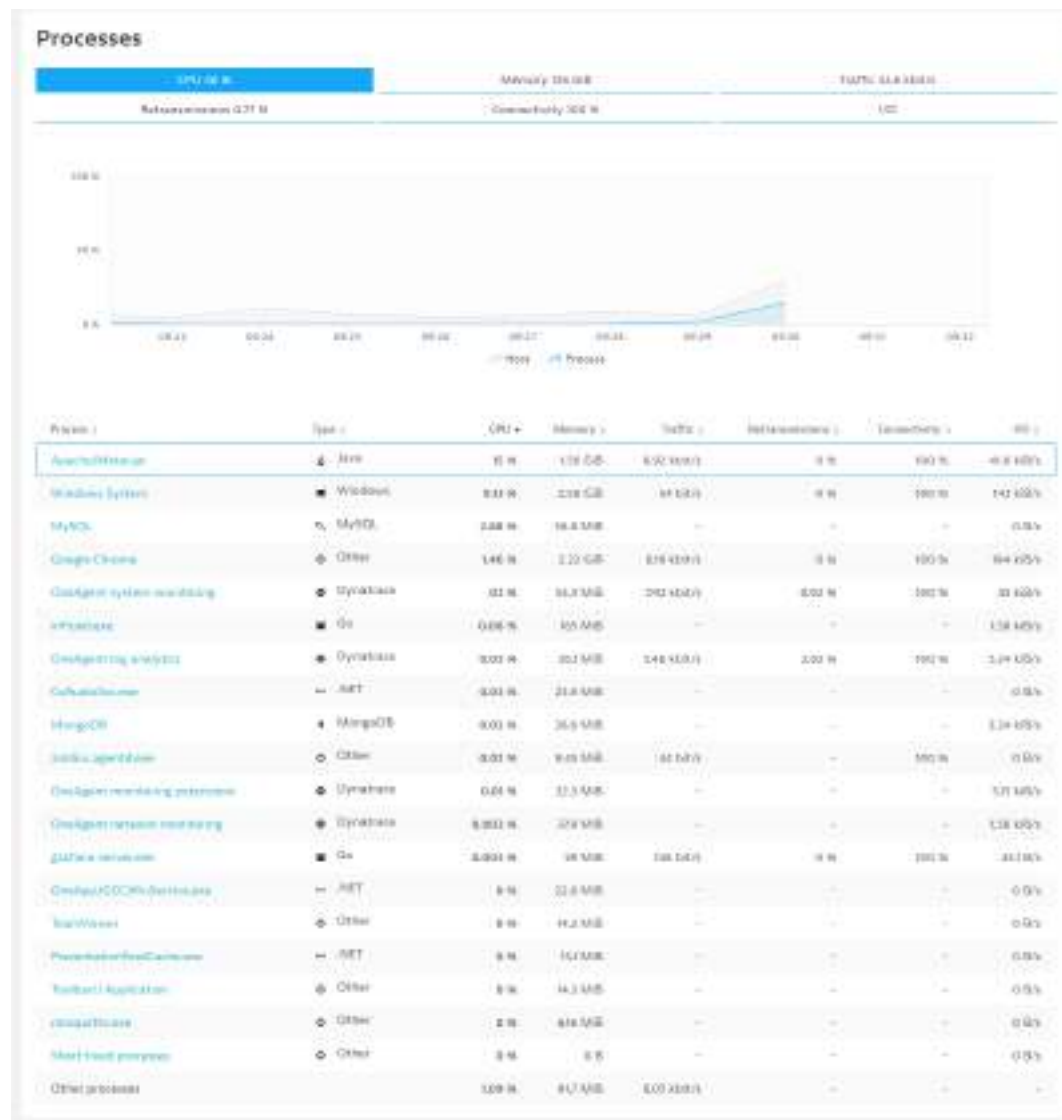
- Adam kukuh kurniawan (2018) ‘Sebagai Backend Iot Data Storage’.
- Andhikha, I.P. (2018) ‘Implementasi dan Analisis Kinerja MySQL Cluster Menggunakan Metode Load Balancing’, *Jurnal Manajemen Informatika*, 8(1), pp. 53–62. Available at: www.dev.MySQL.com.
- Asiminidis, C. (2018) ‘Database Systems Performance Evaluation for Iot Applications’.
- Aziiz, R.N.N., Ichsan, M.H.H. and Arwani, I. (2017) ‘Implementasi Pengiriman Data Sensor LM35 untuk Perbandingan Waktu *Insert* pada Basis Data MySQL dan MongoDB’, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya*, 1(6), pp. 466–475.
- Eyada, M. and Genidy, M.M.E.L. (2020) ‘Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments’, 8. doi:10.1109/ACCESS.2020.3002164.
- Fitri Marisa *et al.* (2022) *MODEL QUERY REALTIME DATABASE UNTUK TRANSAKSI ONLINE*. President University.
- Hidayah, T. and Saptono, H. (2018) ‘PENERAPAN HIGH AVAILABILITY WEB SERVER MENGGUNAKAN NGINX DAN MODSECURITY’.
- Ilmiah, A. *et al.* (2019) ‘IMPLEMENTASI SISTEM MONITORING JARINGAN MENGGUNAKAN ZABBIX PADA PT SUMBER TRIJAYA’, (672015221).
- Makris, A. *et al.* (2019) ‘Performance evaluation of MongoDB and PostgreSQL for spatio-temporal data’, *CEUR Workshop Proceedings*, 2322.
- Makris, A. *et al.* (2021) ‘Correction to: MongoDB Vs PostgreSQL: a comparative study on performance aspects (GeoInformatica, (2020), 10.1007/s10707-020-00407-w)’, *GeoInformatica*, 25(1), pp. 241–242. doi:10.1007/s10707-020-00424-9.
- Patil, M.M. *et al.* (2017) ‘A qualitative analysis of the performance of MongoDB vs MySQL database based on *insertion* and retrieval operations using a web/android application to explore load balancing-Sharding in MongoDB and its advantages’, *Proceedings of the International Conference on IoT in Social*,

- Mobile, Analytics and Cloud, I-SMAC 2017*, (February), pp. 325–330. doi:10.1109/I-SMAC.2017.8058365.
- Putra, Y.Y. and , Oktania Purwaningrum, R.H.W. (2022) ‘PERBANDINGAN PERFORMA RESPON WAKTU KUERI’, 15(1), pp. 39–48.
- Qodarbaskoro, S.Y. and Utam, K. dan E. (2021) ‘PENGUNAAAAN BASIS DATA RELASIONAL DAN NON RELASIONAL PADA APLIKASI IOT (INTERNET OF THINGS)’ , 2(1), pp. 136–144.
- Ronal Watrianthos and Iwan Purnama (2018) *Buku Ajar SISTEM OPERASI*. Uwais Inspirasi Indonesia.
- Suendri, S. (2020) ‘Optimalisasi Sistem Informasi Geografis Bantuan Dana Desa Menggunakan *Database* Cloud Berbasis Dokumen’, *JISTech (Journal of Islamic Science and Technology)*, 5(1), pp. 80–87. Available at: <http://jurnal.uinsu.ac.id/index.php/jistech/article/view/7803>.
- Sulasno, S. and Saleh, R. (2020) ‘Desain dan Implementasi Sistem Monitoring Sumber Daya *Server* Menggunakan Zabbix 4.0’, *JUITA: Jurnal Informatika*, 8(2), p. 187. doi:10.30595/juita.v8i2.6886.
- Wicaksono, N.Y. (2018) ‘PERBANDINGAN KINERJA HBASE DAN MONGODB SEBAGAI BACKEND IOT DATA STORAGE’.
- Fitri Marisa *et al.* (2022) *MODEL QUERY REALTIME DATABASE UNTUK TRANSAKSI ONLINE*. President University.
- Hidayah, T. and Saptono, H. (2018) ‘PENERAPAN HIGH AVAILABILITY WEB SERVER MENGGUNAKAN NGINX DAN MODSECURITY’.
- Ronal Watrianthos and Iwan Purnama (2018) *Buku Ajar SISTEM OPERASI*. Uwais Inspirasi Indonesia.
- WilliamDAssafMSFT (2023) *Memantau dan Memecahkan Masalah Penggunaan Memori - SQL Server*. Available at: <https://learn.microsoft.com/id-id/sql/relational-databases/in-memory-oltp/monitor-and-troubleshoot-memory-usage> (Accessed: 14 June 2023).
- Yonly Adrianus, B., Ade Ria Pratami, M. and Enjel Suryanti, M. (2022) ‘Optimalisasi teamviewer dalam mendukung kegiatan praktikum mahasiswa

dari rumah', *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, 5(1), pp. 144–150.



Lampiran program yang berjalan dikomputer server



Lampiran CPU Usage MySQL data string

Pengujian 1

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std...	Error...	Throughput	Received K...	Sent KB/sec	Avg. Bytes
JDBC Req...	100	10	9	52	4.52	0.00%	93.3/sec	12090.35	0.00	132007.0
TOTAL	100	10	9	52	4.52	0.00%	93.3/sec	12090.35	0.00	132007.0

Pengujian 2

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std...	Error...	Throughput	Received K...	Sent KB/sec	Avg. Bytes
IOBC Req...	100	15	14	75	5.11	0.00%	614/sec	13336.58	0.00	222330.0
TOTAL	100	15	14	75	5.11	0.00%	614/sec	13336.58	0.00	222330.0

Pengujian 3

CPU, Memory dan Disk I/O

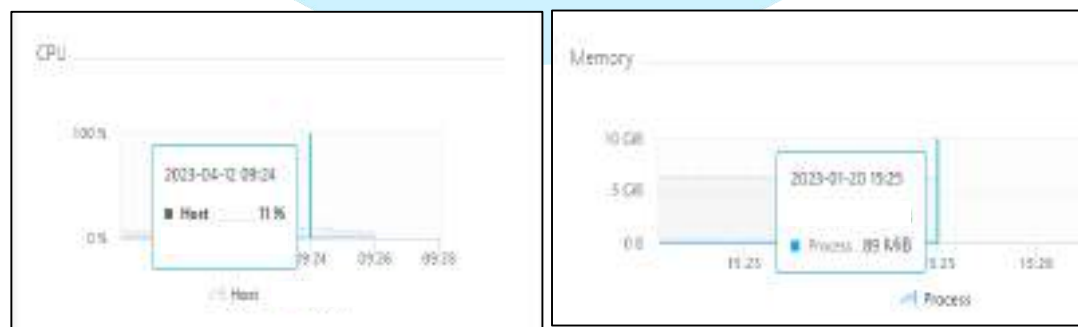


Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std...	Error...	Throughput	Received K...	Sent KB/sec	Avg. Bytes
IOBC Req...	100	24	3	1807	179...	0.00%	40.0/sec	1794.39	0.00	45548.0
TOTAL	100	24	3	1807	179...	0.00%	40.0/sec	1794.39	0.00	45548.0

Pengujian 4

CPU, Memory dan Disk I/O



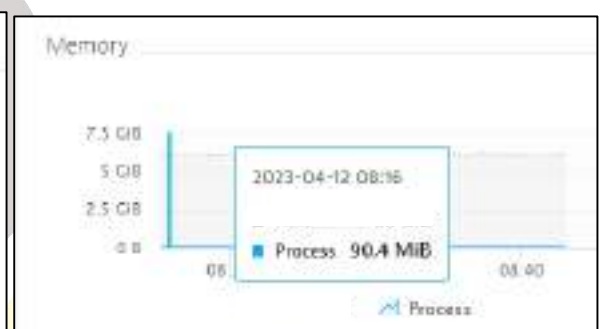
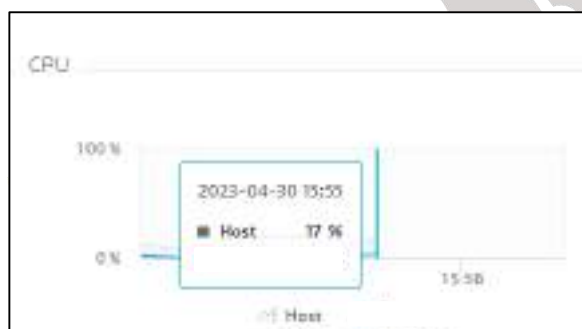


Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std.	Error	Throughput	Received K...	Sent KB/sec	Av. Bytes
JDBC Req.	100	30	28	41	3.03	0.00%	32.6/sec	14211.97	0.00	447068.0
TOTAL	100	30	28	41	3.03	0.00%	32.6/sec	14211.01	0.00	447068.0

Pengujian 5

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std.	Error	Throughput	Received K...	Sent KB/sec	Av. Bytes
JDBC Req.	100	45	42	78	4.75	0.00%	21.5/sec	14402.03	0.00	674354.0
TOTAL	100	45	42	78	4.75	0.00%	21.5/sec	14402.03	0.00	674354.0

Pengujian 6

CPU, Memory dan Disk I/O



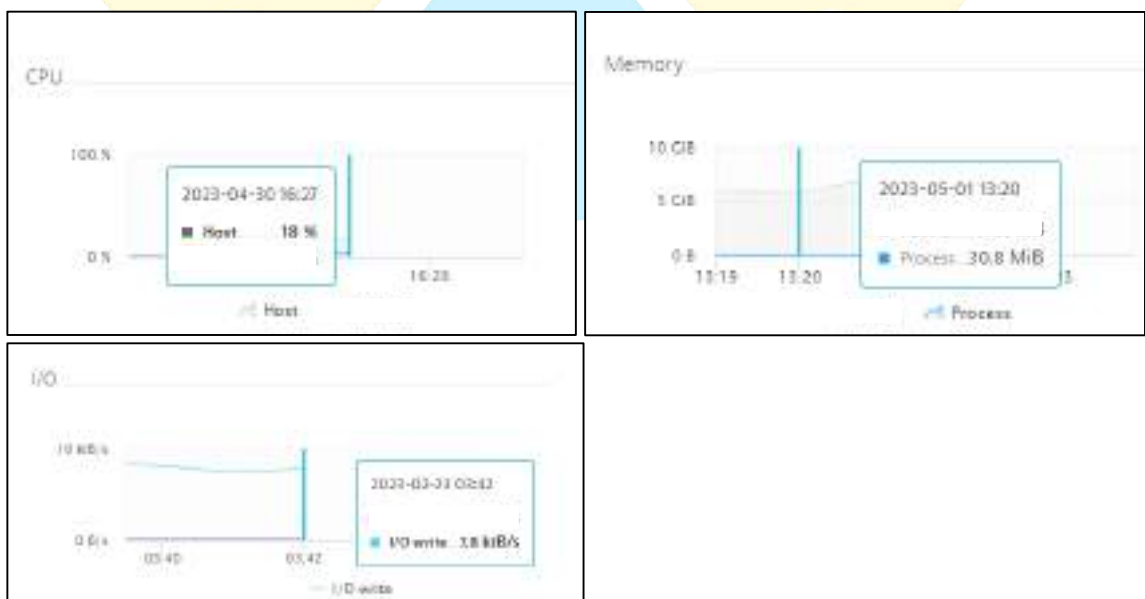
Runtime dan Throughput

Label	#Samples	Average	Min	Max	Std...	Error...	Throughput	Received K...	Sent KB/sec	Avg. Bytes
JDBC Req...	100	62	56	91	6.70	0.00%	15.9/sec	14055.20	0.00	902561.0
TOTAL	100	62	56	91	6.70	0.00%	15.9/sec	14055.20	0.00	902561.0

Lampiran CPU Usage MongoDB data string

Pengujian 1

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received...	Sent KB/s	Avg. Bytes
JSR223 re...	100	65	31	2001	195.99	0.00%	15.3/sec	0.00	0.00	0
TOTAL	100	65	31	2001	195.99	0.00%	15.3/sec	0.00	0.00	0

Pengujian 2

CPU, Memory dan Disk I/O



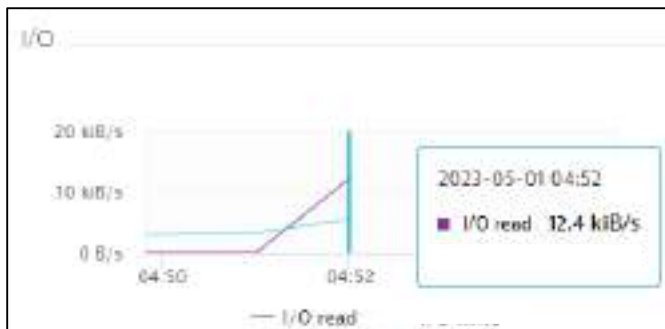
Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received...	Sent KB/s	Avg. Bytes
JSR223 re...	100	104	32	315	39.08	0.00%	9.6/sec	0.00	0.00	0
TOTAL	100	104	32	315	39.08	0.00%	9.6/sec	0.00	0.00	0

Pengujian 3

CPU, Memory dan Disk I/O





Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received	Sent KB/s	Avg. Bytes
GR223 re...	100	163	134	357	90.66	0.00%	6.1/sec	0.00	0.00	0
TOTAL	100	163	134	357	90.66	0.00%	6.1/sec	0.00	0.00	0

Pengujian 4

CPU, Memory dan Disk I/O

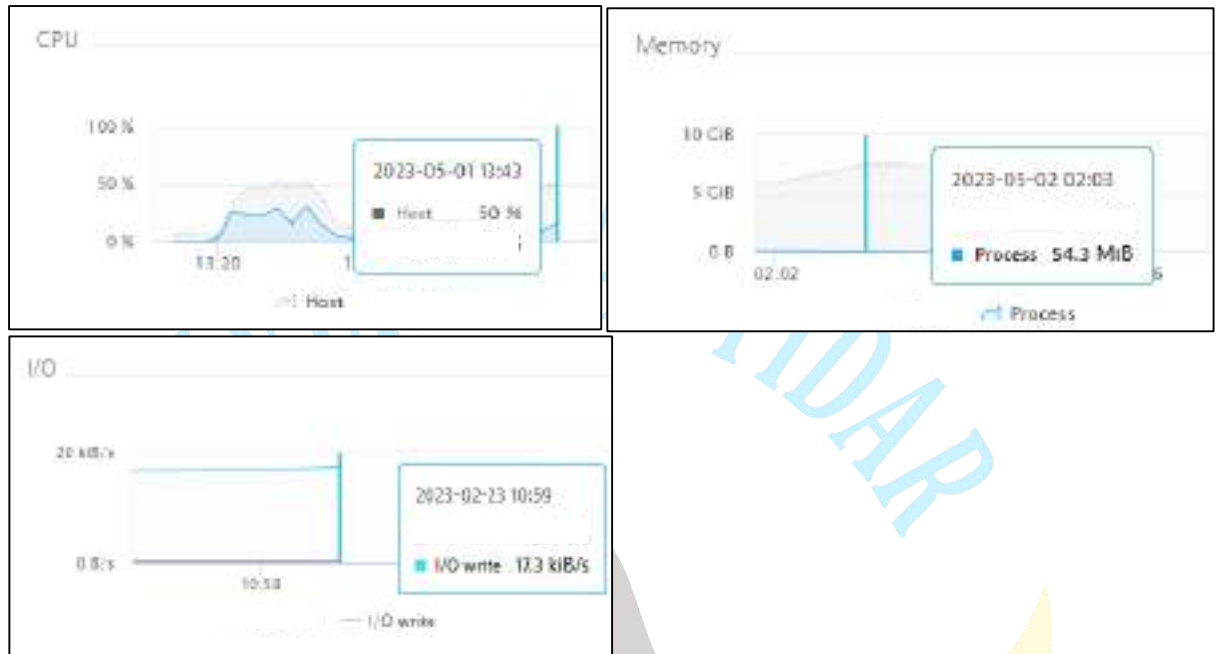


Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received	Sent KB/s	Avg. Bytes
ISB223 re...	100	433	273	1752	188.58	0.00%	2.3/sec	0.00	0.00	0
TOTAL	100	433	273	1752	188.58	0.00%	2.3/sec	0.00	0.00	0

Pengujian 5

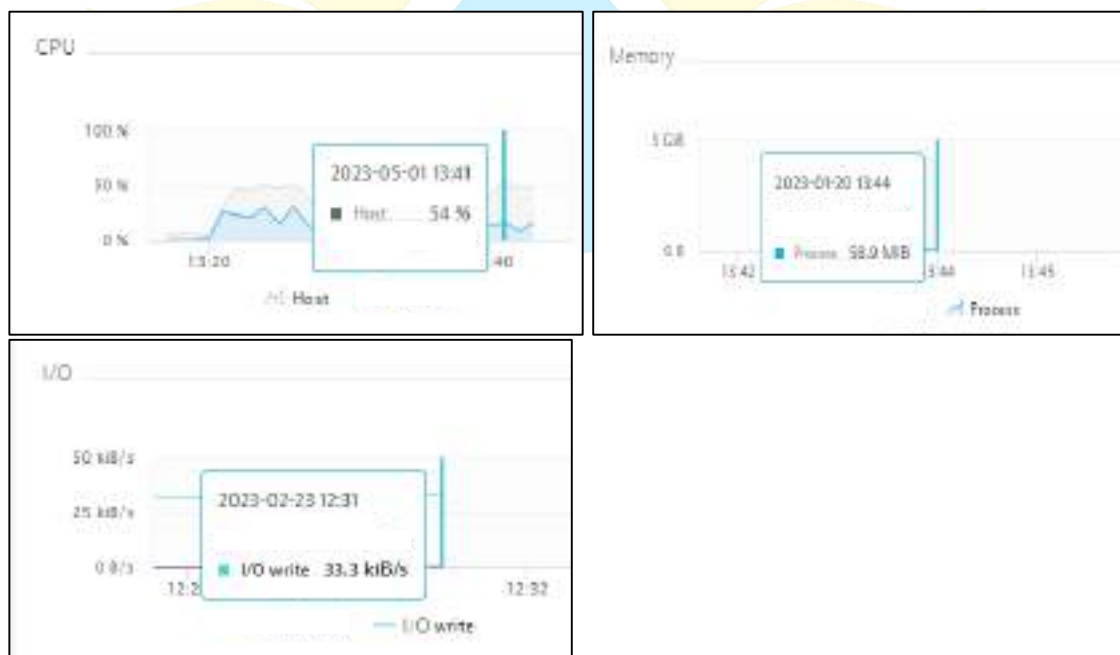
CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received	Sent KB/s	Avg. Bytes
SR23 re	100	797	440	3179	423.34	0.00%	1.3/sec	0.00	0.00	0
TOTAL	100	797	440	3179	423.34	0.00%	1.3/sec	0.00	0.00	0

Pengujian 6



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received	Sent KB/s	Avg Bytes
JSP222 re...	100	1003	560	9214	1038.26	0.03%	49.8/min	0.00	0.00	0
TOTAL	100	1003	560	9214	1038.26	0.03%	49.8/min	0.00	0.00	0

Lampiran CPU Usage InfluxDB data string

Pengujian 1

CPU, Memory dan Disk I/O



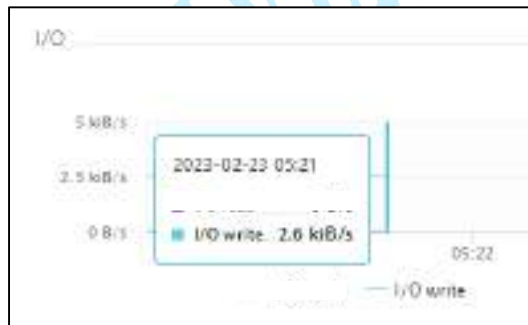
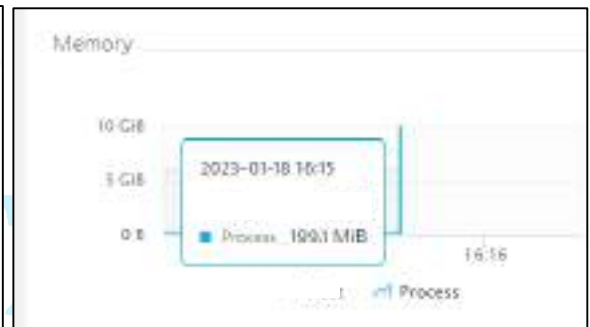
Pengujian 2

CPU, Memory dan Disk I/O



Pengujian 3

CPU, Memory dan Disk I/O

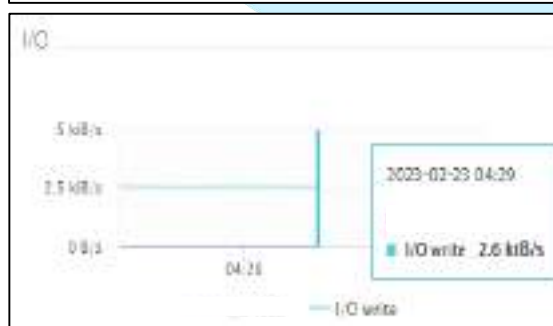
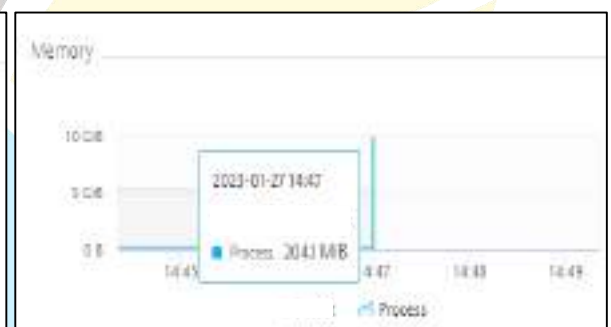
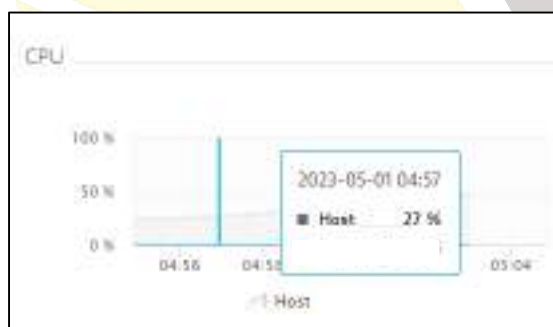


Runtime dan Throughput

Stats	
Total request time	11.7 s
Data processing time	0.200 ms
Number of queries	100

Pengujian 4

CPU, Memory dan Disk I/O



Runtime dan Throughput

Stats	
Total request time	13.9 s
Data processing time	0.1000 ms
Number of queries	99

Pengujian 5

CPU, Memory dan Disk I/O



Runtime dan Throughput

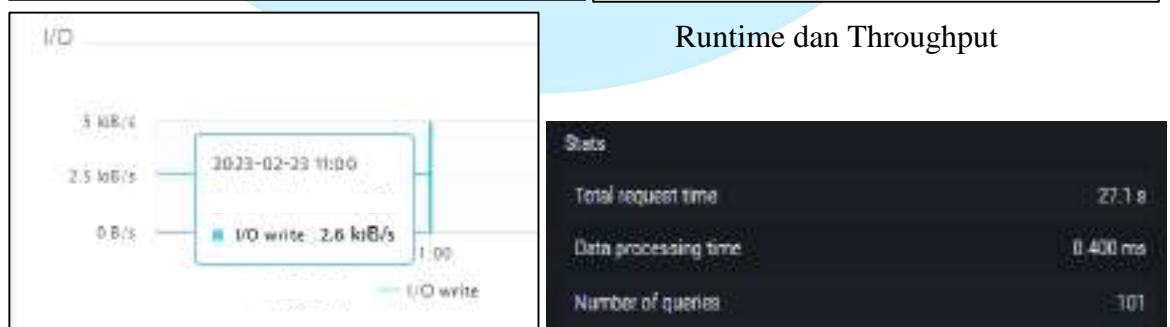


Pengujian 6

CPU, Memory dan Disk I/O



Runtime dan Throughput



Lampiran CPU Usage MySQL data gambar

Pengujian 1

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received	Sent KB/s	Avg. Bytes
DBC Req...	28	165	51	1065	190.32	0.09%	6.0/sec	34415.37	0.00	5053966.0
TOTAL	28	165	51	1065	190.32	0.09%	6.0/sec	34415.37	0.00	5053966.0

Pengujian 2

CPU, Memory dan Disk I/O





Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received	Sent KiB/s	Avg. Bytes
KCRC Req.	20	261	173	265	22.98	0.00%	49/sec	93171.66	0.00	14281954.0
TOTAL	20	261	173	265	22.98	0.00%	49/sec	93171.66	0.00	14281954.0

Pengujian 3

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received	Sent KiB/s	Avg. Bytes
KCRC Req.	20	349	292	408	43.31	0.00%	2.9/sec	93557.44	0.00	33511826.0
TOTAL	20	349	292	408	43.31	0.00%	2.9/sec	93557.44	0.00	33511826.0

Pengujian 4

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	#Samples	Average	Min	Max	Std.Dev	Error %	Throughput	Received	Sent KB/s	Avg. Bytes
DBC Req...	20	673	547	1207	139.27	0.00%	1.5/sec	87668.01	0.00	60479427.0
TOTAL	20	673	547	1207	139.27	0.00%	1.5/sec	87668.01	0.00	60479427.0

Pengujian 5

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughp...	Received...	Sent KB/s...	Avg. Bytes
DBC Req...	20	1876	1508	3973	568.42	0.00%	32.0/min	51945.85	0.00	94831787.0
TOTAL	20	1876	1508	3973	568.42	0.00%	32.0/min	51945.85	0.00	94831787.0

Pengujian 6

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughp...	Received...	Sent KB/s...	Avg. Bytes
DBC Req...	20	6677	2584	13981	2516.28	0.00%	90/min	21115.36	0.00	1444227...
TOTAL	20	6677	2584	13981	2516.28	0.00%	90/min	21115.36	0.00	1444227...

Lampiran CPU Usage MongoDB data gambar

Pengujian 1

CPU, Memory dan Disk I/O





Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std.Dev	Error %	Throughput	Received..	Sent K2/sec	Avg.Bytes
JSR223 read	10	802	183	2653	870.59	0.00%	1.2/sec	0.00	0.00	0
TOTAL	10	802	183	2653	870.59	0.00%	1.2/sec	0.00	0.00	0

Pengujian 2

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std.Dev	Error %	Throughput	Received..	Sent K2/sec	Avg.Bytes
JSR223 read	10	994	364	2271	470.64	0.00%	1.1/sec	0.00	0.00	0
TOTAL	10	994	364	2271	470.64	0.00%	1.1/sec	0.00	0.00	0

Pengujian 3

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received	Sent KB/sec	Avg. Bytes
JSR223 read	10	204	942	6769	1778.82	0.00%	26.6/min	0.00	0.00	0
TOTAL	10	204	942	6769	1778.82	0.00%	26.6/min	0.00	0.00	0

Pengujian 4

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received	Sent KB/sec	Avg. Bytes
JSR223 re...	10	3676	2945	4151	370.38	0.00%	16.3/min	0.00	0.00	0
TOTAL	10	3676	2945	4151	370.38	0.00%	16.3/min	0.00	0.00	0

Pengujian 5

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received	Sent KB/s	Avg. Bytes
ISR223 re...	10	4381	3428	8057	1277.49	0.00%	13.7/min	0.00	0.00	.0
TOTAL	10	4381	3428	8057	1277.49	0.00%	13.7/min	0.00	0.00	.0

Pengujian 6

CPU, Memory dan Disk I/O



Runtime dan Throughput

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received	Sent KB/s	Avg. Bytes
ISR223 re...	10	4753	4268	9559	388.46	0.00%	12.6/min	0.00	0.00	.0
TOTAL	10	4753	4268	9559	388.46	0.00%	12.6/min	0.00	0.00	.0

Lampiran CPU Usage InfluxDB data gambar

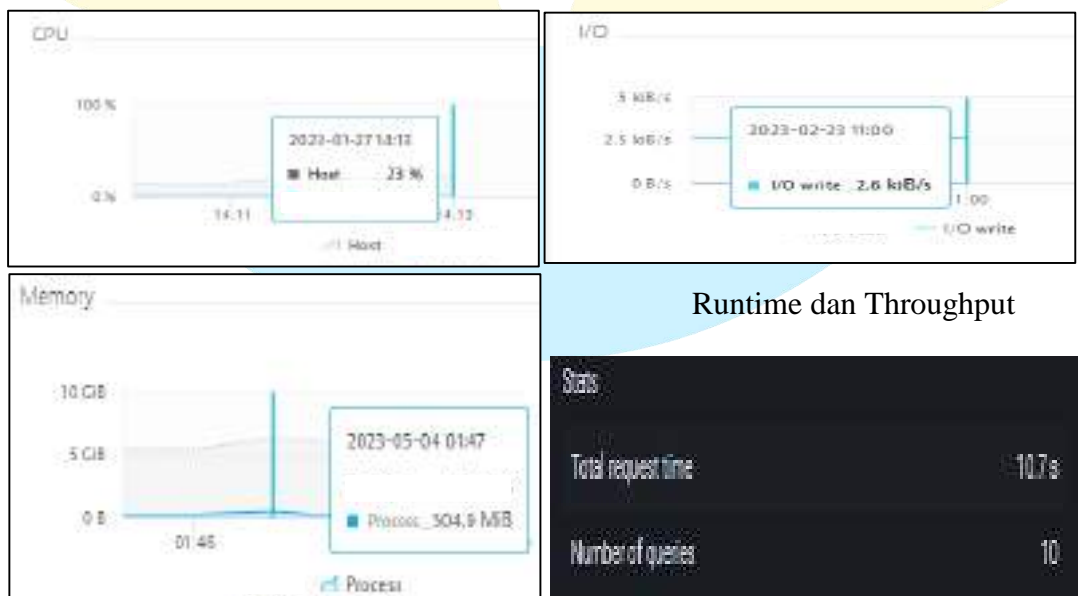
Pengujian 1

CPU, Memory dan Disk I/O



Pengujian 2

CPU, Memory dan Disk I/O



Pengujian 3

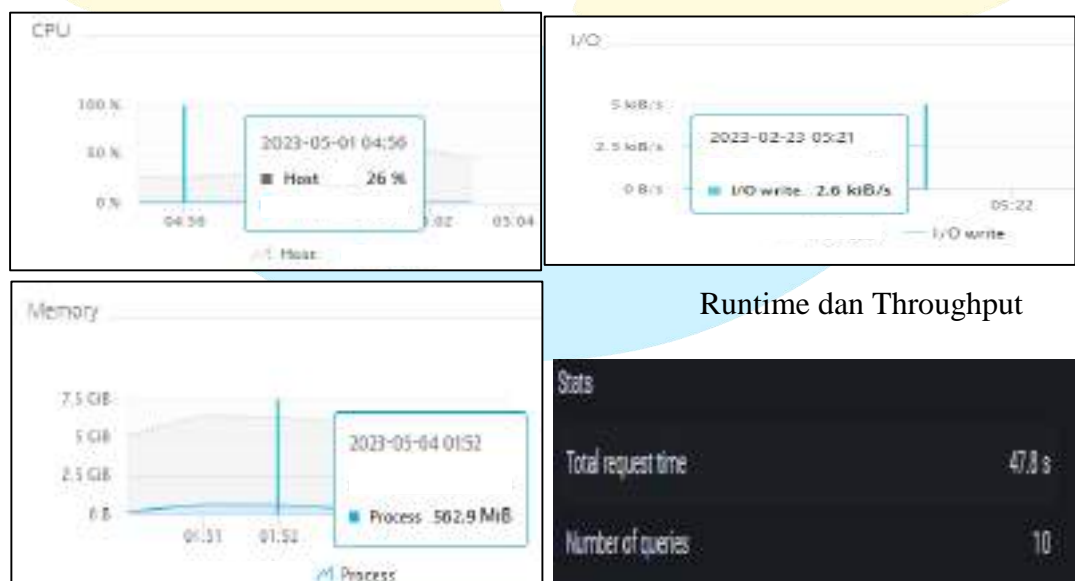
CPU, Memory dan Disk I/O



Runtime dan Throughput

Pengujian 4

CPU, Memory dan Disk I/O



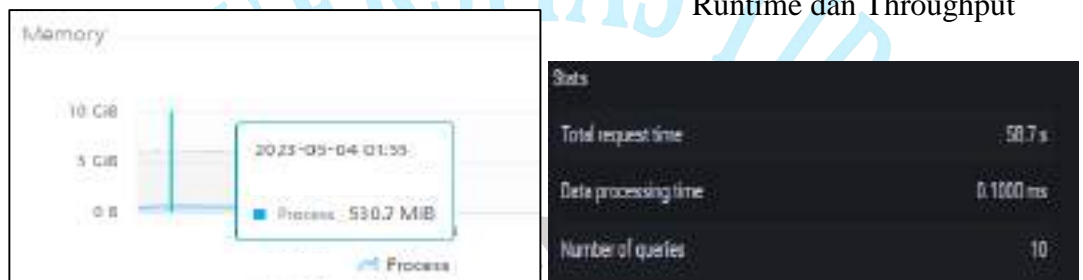
Runtime dan Throughput

Pengujian 5

CPU, Memory dan Disk I/O



Runtime dan Throughput

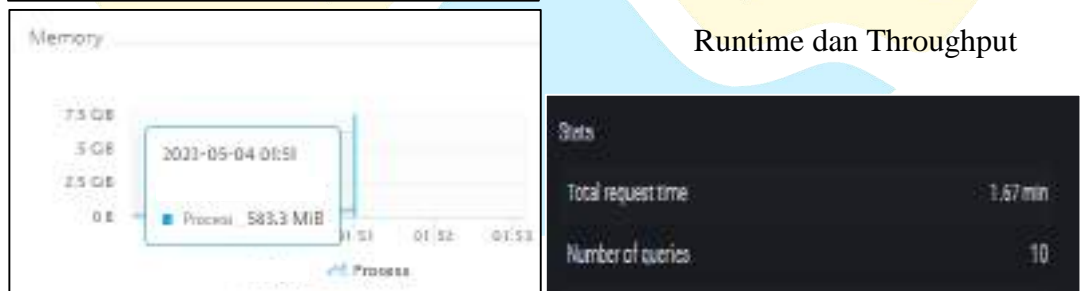


Pengujian 6

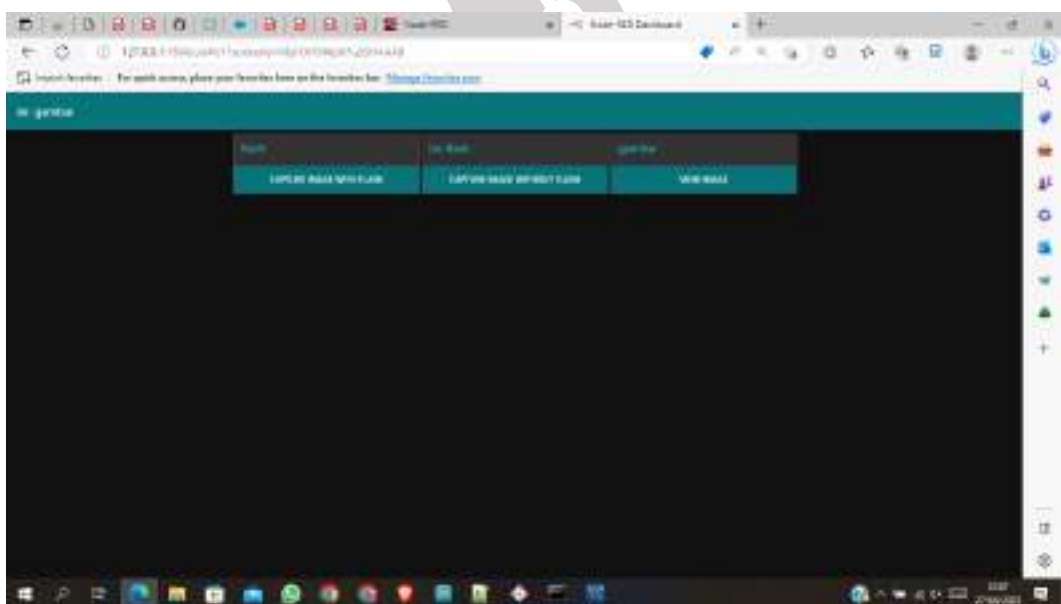
CPU, Memory dan Disk I/O



Runtime dan Throughput



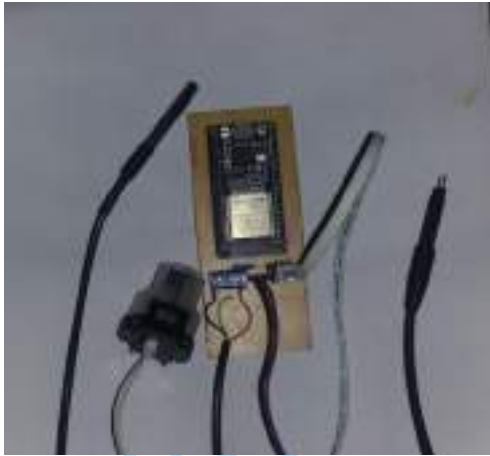
Tampilan web *IoT*



Proses pembuatan alat



Esp32 dan sensor



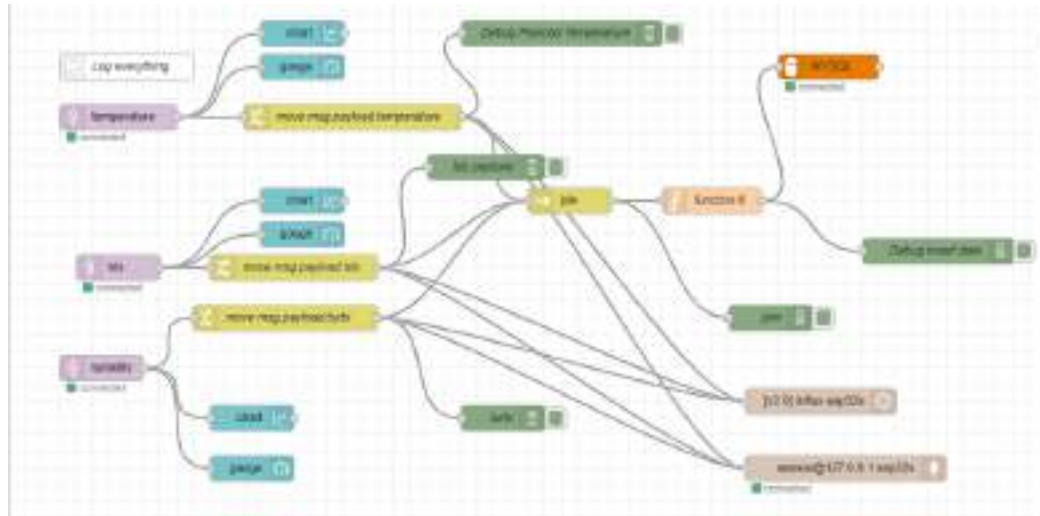
Esp32cam



Box atau casing alat



Flow node-red



Query MySQL

```
CREATE TABLE `sensor` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `suhu` float NOT NULL,
  `tds` float NOT NULL,
  `kekeruhan` float NOT NULL,
  `created_at` TIMESTAMP NOT NULL DEFAULT current_timestamp,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
```

```
esp32  
function("sensor")
```

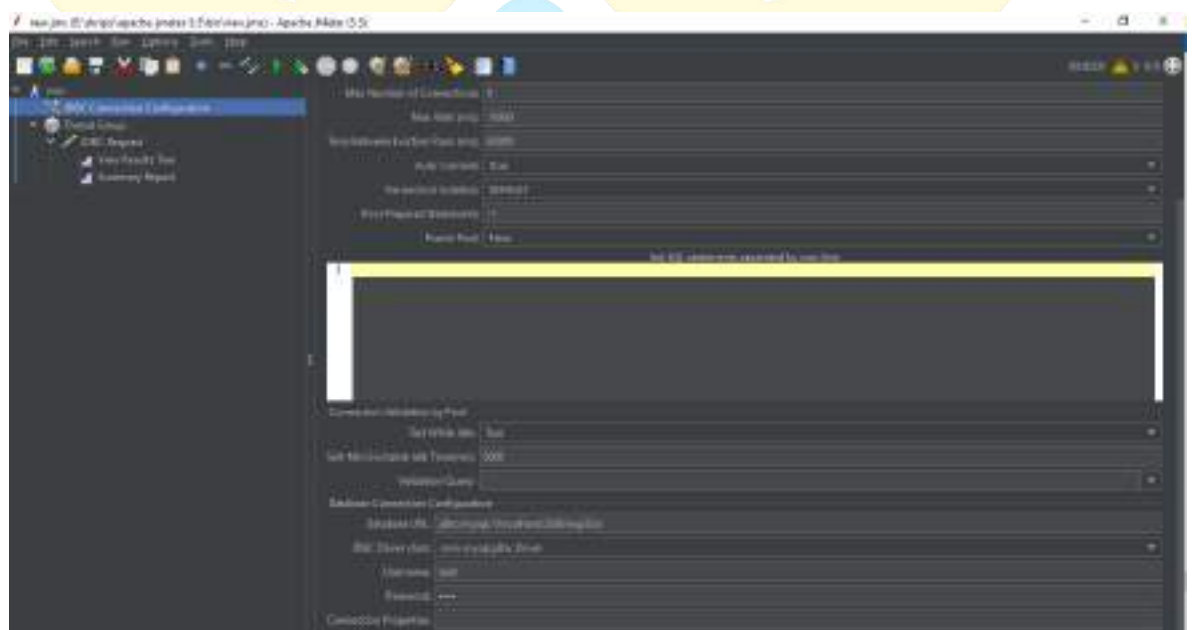
Query MongoDB

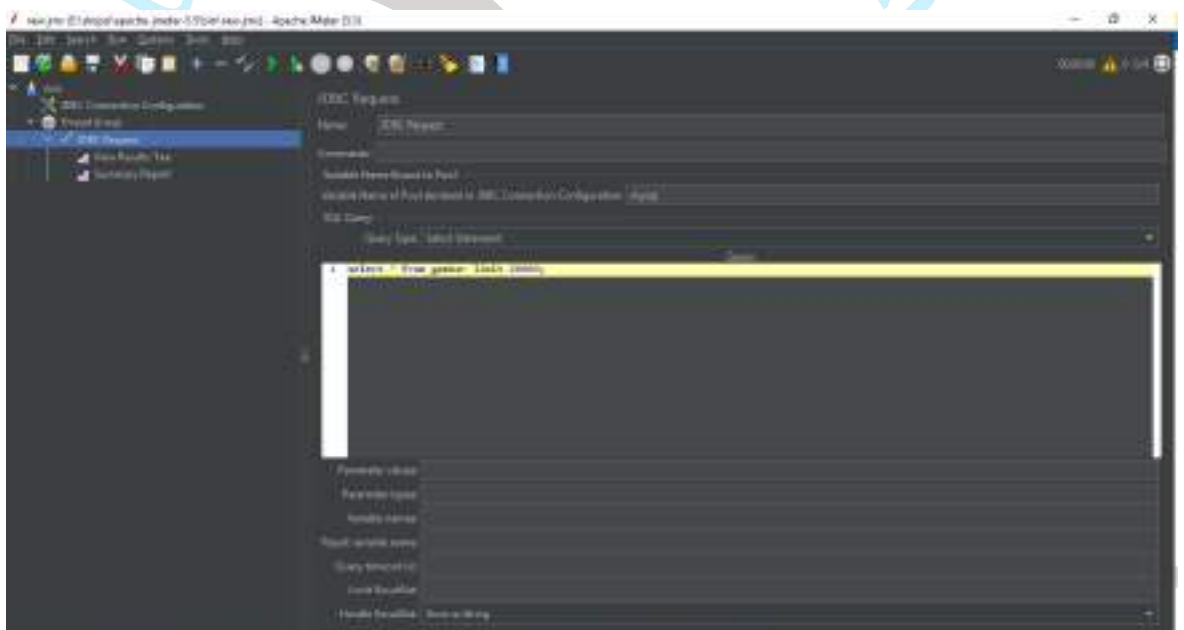
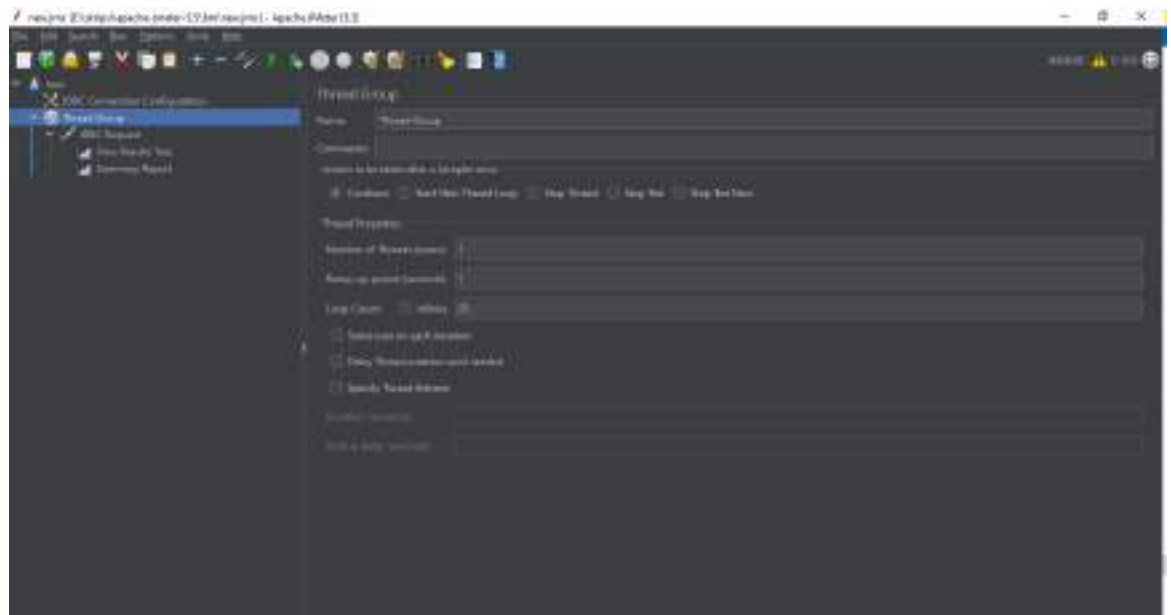
```
> use esp32
switched to db esp32
> db.createCollection("sensor")
{ "ok" : 1 }
>
```

Query InfluxDB

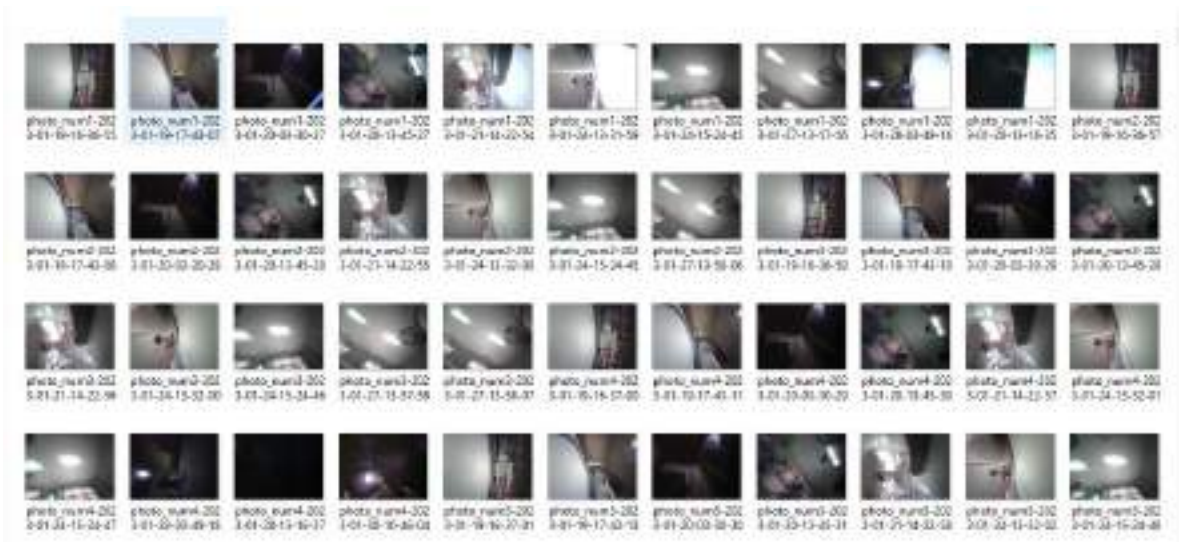
```
1 from(bucket: "sensor")
2 |> range(start: v.timeRangeStart, stop:v.timeRangeStop)
3 |> filter(fn: (r) => r._measurement == "esp32s")
4 |> filter(fn: (r) => r._field == "tds" or r._field == "temperature" or r._field == "turbid")
5 |> limit(n: 20000)
6
7 from(bucket: "images")
8 |> range(start: v.timeRangeStart, stop:v.timeRangeStop)
9 |> filter(fn: (r) => r._measurement == "esp")
10 |> limit(n: 1000)
```

Konfigurasi apache jmeter

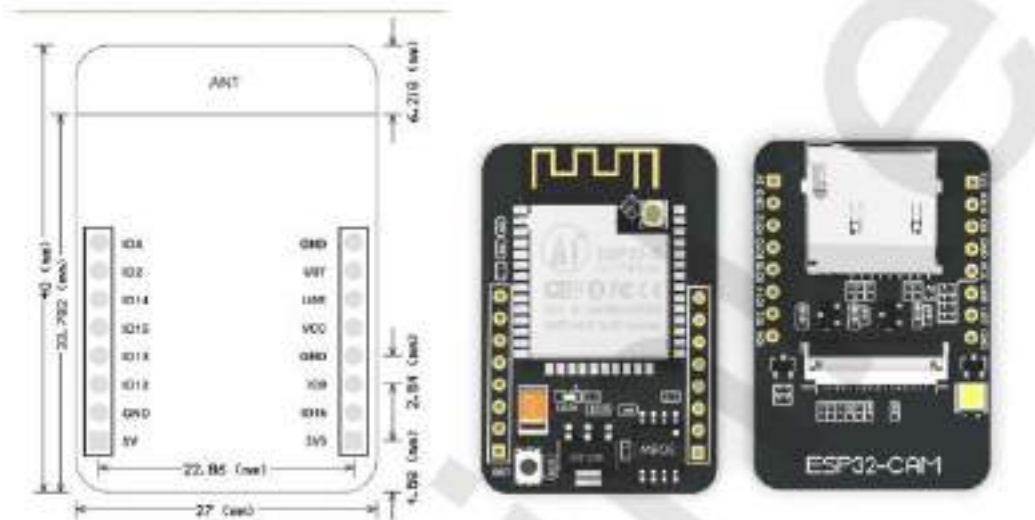




Sampel file gambar



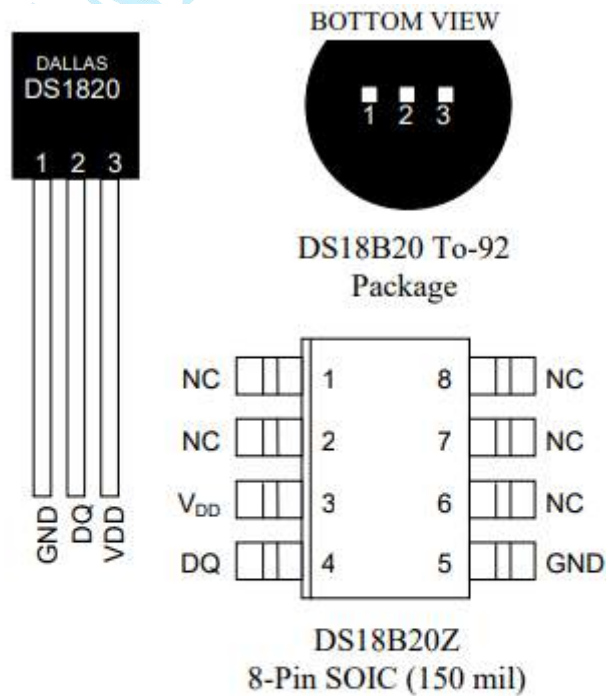
Esp32-Cam Module



CAM	ESP32	SD	ESP32
D0	PIN5	CLK	PIN14
D1	PIN18	CMD	PIN15
D2	PIN19	DATA0	PIN2
D3	PIN21	DATA1/FLASH LAMP	PIN4
D4	PIN36	DATA2	PIN12
D5	PIN39	DATA3	PIN13
D6	PIN34		

CAM	ESP32	SD	ESP32
D7	PIN35		
XCLK	PIN0		
PCLK	PIN22		
VSYNC	PIN25		
HREF	PIN23		
SDA	PIN26		
SCL	PIN27		
POWER PIN	PIN32		

Sensor DS18B20

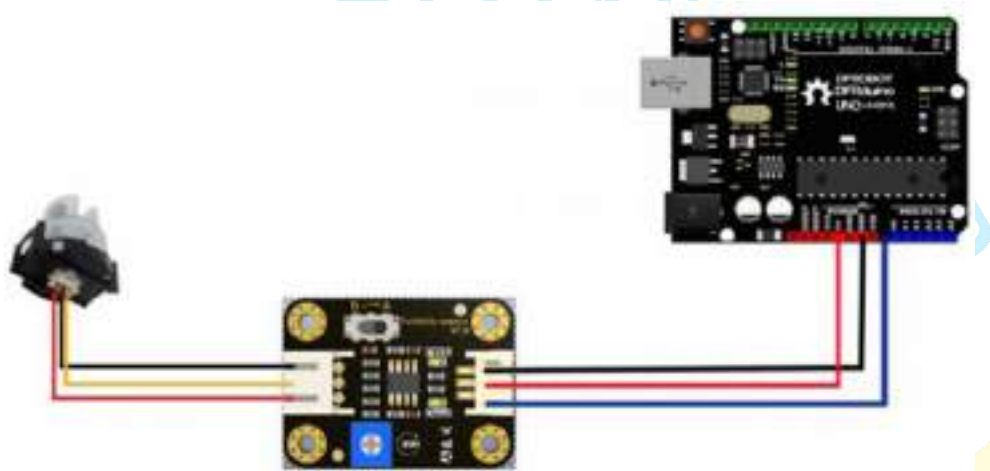


PIN DESCRIPTION

GND - Ground
DQ - Data In/Out
V_{DD} - Power Supply Voltage
NC - No Connect

PIN 8PIN SOIC	PIN TO92	SYMBOL	DESCRIPTION
5	1	GND	Ground.
4	2	DQ	Data Input/Output pin. For 1-Wire operation: Open drain. (See "Parasite Power" section.)
3	3	V _{DD}	Optional V_{DD} pin. See "Parasite Power" section for details of connection. V _{DD} must be grounded for operation in parasite power mode.

Turbidity sensor SKU: SEN0189



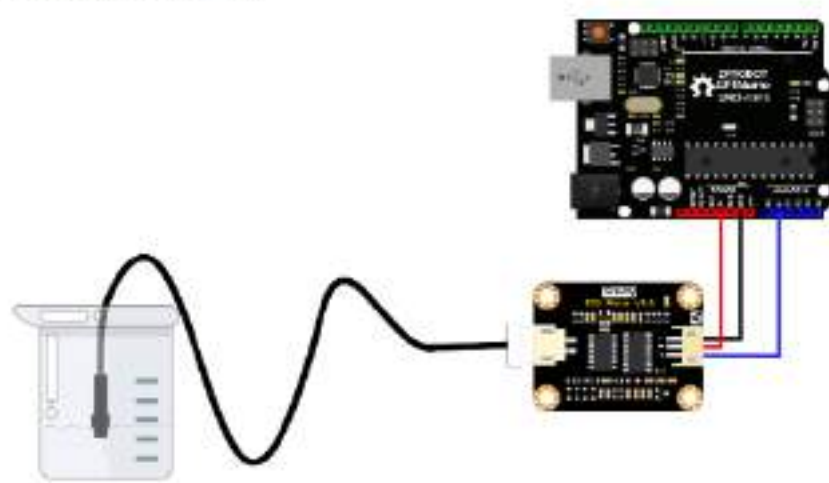
1. "D/A" Output Signal Switch
2. "A": Analog Signal Output, the output value will decrease when in liquids with a high turbidity
3. "D": Digital Signal Output, high and low levels, which can be adjusted by the threshold potentiometer
4. Threshold Potentiometer: you can change the trigger condition by adjusting the threshold potentiometer in digital signal mode.

Specification

- Operating Voltage: 5V DC
- Operating Current: 40mA (MAX)
- Response Time : <500ms
- Insulation Resistance: 100M (Min)
- Output Method:
Analog output: 0-4.5V
Digital Output: High/Low level signal (you can adjust the threshold value by adjusting the potentiometer)
- Operating Temperature: 5°C~90°C
- Storage Temperature: -10°C~90°C
- Weight: 30g
- Adapter Dimensions: 38mm*28mm*10mm/1.5inches *1.1inches*0.4inches

Sensor TDS

Connection Diagram



Num	Label	Description
1	-	Power GND(0V)
2	+	Power VCC(3.3 ~ 5.5V)
3	A	Analog Signal Output(0 ~ 2.3V)
4	TDS	TDS Probe Connector
5	LED	Power Indicator

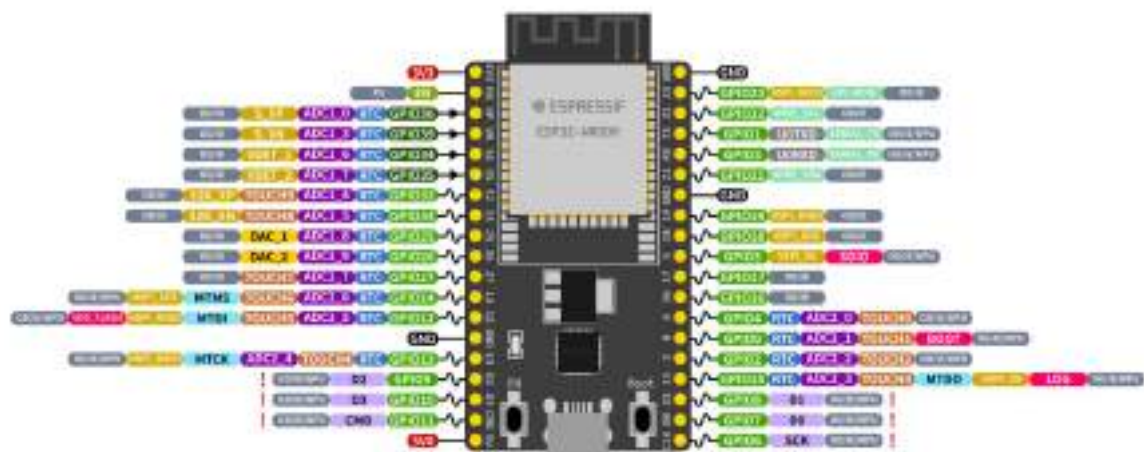
Specification

- Signal Transmitter Board
 - Input Voltage: 3.3 ~ 5.5V
 - Output Voltage: 0 ~ 2.3V
 - Working Current: 3 ~ 6mA
 - TDS Measurement Range: 0 ~ 1000ppm
 - TDS Measurement Accuracy: $\pm 10\%$ F.S. (25 °C)
 - Module Size: 42 * 32mm
 - Module Interface: PH2.0-3P
 - Electrode Interface: XH2.54-2P
- TDS probe
 - Number of Needle: 2
 - Total Length: 83cm
 - Connection Interface: XH2.54-2P
 - Colour: Black
 - Other: Waterproof Probe

PIN Esp32

ESP32-DevKitC

ESPRESSIF



ESP32 Specs

32-bit Xtensa dual-core @240MHz
Wi-Fi IEEE 802.11 b/g/n 2.4GHz
Bluetooth 4.2 BR/EDR and BLE
528-KB SRAM (16-KB for co-processor)
448-KB ROM
34 GPIOs, 4x SPI, 3x UART, 2x I2C,
2x I2S, RMT, LED PWM, 1 host SD/eMMC/SDIO,
1 slave SDIO/SPI, TWAI, 12-bit ADC, Ethernet

