

Dashboard Pemantauan Integrasi Pembangkit Energi Terbarukan di Kampus ITS

Tony David Gultom, Wirawan, dan Dedet Candra Riawan.
Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: wirawan@ee.its.ac.id

Abstrak—Listrik merupakan komponen penting dalam kehidupan dikarenakan hampir semua kegiatan membutuhkan energi listrik. Kebutuhan akan listrik konvensional yang semakin meningkat dapat mengakibatkan krisis listrik yang terjadi kapan saja. Untuk memenuhi kebutuhan tersebut, dibutuhkan energi alternatif yang dapat membantu mendapatkan suplai energi listrik. Pada kampus ITS sudah terpasang 4 Pembangkit Listrik Tenaga Surya (PLTS) yang digunakan untuk membantu penyediaan listrik di kampus. Keempat PLTS ini memiliki sistem monitoring yang berfungsi untuk membaca variabel yang dihasilkan dari keempat PLTS yang telah terpasang. Namun, keempat sistem monitoring terbatas pada penyediaan data sehingga menghambat pengguna dalam melakukan pembacaan data pada PLTS tersebut. Oleh karena itu, penelitian ini dilakukan dengan tujuan untuk merancang sebuah sistem *monitoring* baru berupa *dashboard* yang dapat melakukan pembacaan terhadap variabel dari keempat PLTS berbasis *Internet of Things (IoT)*. Dalam penelitian ini akan dirancang sebuah *dashboard* yang akan melakukan pembacaan variabel dari keempat PLTS di kampus ITS. *Dashboard* ini juga akan memiliki fungsi berupa pencatatan (*logging*) dan juga mampu untuk menampilkan Analisa data dari hasil *logging*. Sistem ini juga bersifat *open-source* yang artinya dapat memberikan ruang untuk dikembangkan di kemudian hari.

Kata Kunci—*Dashboard, Internet of Things, Monitoring, Open Source, Pembangkit Listrik Tenaga Surya (PLTS)*.

I. PENDAHULUAN

LISTRIK merupakan komponen penting dalam kehidupan dikarenakan hampir semua rumah menggunakan energi listrik dan listrik digunakan dalam berbagai macam kegiatan. Permintaan listrik akan meningkat setiap tahun seiring dengan pertumbuhan populasi, peningkatan tingkat kemakmuran, dan pertumbuhan ekonomi. Kebutuhan akan listrik konvensional yang semakin meningkat dapat mengakibatkan krisis listrik yang terjadi kapan saja. Untuk memenuhi kebutuhan tersebut, dibutuhkan energi alternatif yang dapat membantu mendapatkan suplai energi listrik.

Terdapat berbagai energi alternatif yang dapat dimanfaatkan, salah satunya adalah energi dari cahaya matahari yang dihasilkan dari Pembangkit Listrik Tenaga Surya (PLTS). Pembangkit listrik tenaga surya adalah pembangkit listrik yang memanfaatkan energi dari cahaya matahari untuk menghasilkan energi listrik. Komponen utama dari pembangkit ini adalah panel surya yang dapat mengkonversi energi matahari menjadi energi listrik sehingga dapat digunakan untuk kebutuhan listrik sehari-hari. Besar *output* daya yang dihasilkan dari konversi panel surya ditentukan berdasarkan kondisi lingkungan tempat panel surya tersebut diletakkan contohnya seperti kondisi intensitas cahaya matahari, suhu, arah datangnya matahari, dan spektrum cahaya matahari. Adanya kondisi lingkungan yang

Tabel 1.

Kebutuhan Perangkat Keras

Perangkat Keras	Keterangan
PC	Sebagai media terinstallnya database dan server yang akan digunakan
Inverter yang mendukung Modbus TCP	Agar dapat dilakukan pembacaan pada data yang dihasilkan dari inverter pada jaringan Modbus terhubung

Tabel 2.

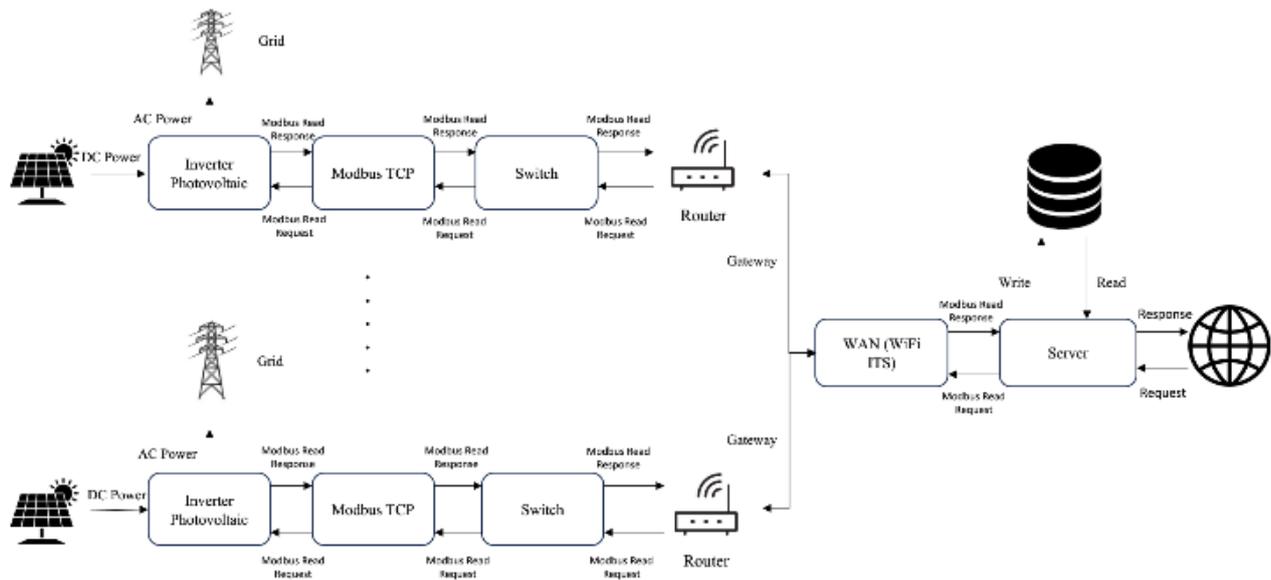
Kebutuhan Perangkat Lunak

Perangkat Lunak	Keterangan
HTML	Sebagai kerangka dalam menampilkan konten pada website
CSS	Untuk memberikan <i>styling</i> pada web
JavaScript	Untuk menambah interaktif pada web dan sebagai bahasa pemrograman pada web dan server
NodeJS	<i>Environment</i> untuk mengembangkan server
PM2	Sebagai <i>process manager</i> yang berfungsi untuk mengatur aktivitas dari server
React	Framework dari JavaScript untuk membuat sebuah <i>user interface</i>
MongoDB	Sebagai database untuk menyimpan data
MongoDB Compass	Sebagai <i>user interface</i> untuk mengakses database dan melihat isi dari database
Visual Studio Code	Aplikasi yang membantu pemrogram mengembangkan kode perangkat lunak secara efisien.

berbeda-beda ini mengakibatkan fluktuasi pada *output* daya yang dihasilkan.

Dalam pengoperasian PLTS, diperlukan adanya sistem *monitoring* yang dapat memantau aktivitas konversi energi matahari menjadi energi listrik dan kinerja dari pembangkit. Sistem *monitoring* yang dibuat dapat memperoleh informasi dari PLTS secara *real-time* dan dapat diakses melalui web sehingga meningkatkan mobilitas dari sistem monitoring ini. Sebelum adanya penelitian ini, sistem *monitoring* pengoperasian PLTS di kampus ITS masih menggunakan *dashboard monitoring* bawaan dari perangkat yang dipasang pada pembangkit. *Dashboard* ini dapat diakses melalui internet namun terbatas pada penyajian data tertentu sehingga membuat terbatasnya ruang dalam pengembangan sistem *monitoring*. Percobaan untuk pembuatan sistem *monitoring* sudah pernah dilakukan menggunakan *Human Machine Interface (HMI)* di ruangan AJ301, namun hal ini membatasi mobilitas dari sistem karena *monitoring* harus dilakukan di tempat HMI terpasang.

Berdasarkan latar belakang permasalahan, penelitian bertujuan untuk memberikan sistem *monitoring* PLTS baru berupa *dashboard* berbasis web yang dapat mengakses data



Gambar 1. Desain Sistem *Monitoring*.

dari PLTS secara *real-time*. *Dashboard* ini juga dapat memberikan ruang untuk pengembangan sistem *monitoring* pada bagian penyajian data kedepannya.

II. TINJAUAN PUSTAKA

A. *Internet of Things*

Internet of Things (IoT) dapat diartikan sebagai kombinasi dari sensor dan aktuator yang menyediakan dan menerima informasi yang didigitalkan dan ditempatkan ke dalam jaringan dua arah untuk mengirimkan semua data untuk digunakan oleh banyak layanan dan pengguna. Beberapa sensor dapat dipasang pada sebuah perangkat untuk mengukur variabel pada perangkat dan kemudian mengirimkan data tersebut ke *cloud*. Arsitektur sistem IoT dapat dibagi menjadi empat lapisan (*layer*) antara lain:

1. *Object sensing layer*

Memperoleh data dari perangkat.

2. *Data exchange layer*

Transmisi data melalui jaringan komunikasi.

3. *Information integration layer*

Memproses informasi yang diperoleh dari jaringan, memfilter data yang tidak diinginkan dan memberikan informasi utama menjadi data yang dapat digunakan sebagai layanan atau untuk pengguna.

4. *Application service layer*

Menyediakan layanan konten kepada pengguna.

Beberapa keuntungan dan manfaat dari IoT adalah berkembangnya inovasi layanan yang dapat meningkatkan kinerja dan pengurangan biaya [1].

B. *Pembangkit Listrik Tenaga Surya*

Pembangkit Listrik Tenaga Surya adalah pembangkit listrik yang mengubah cahaya matahari menjadi listrik. PLTS sering juga disebut sebagai *Solar Cell* atau *Solar Photovoltaic*. *Photovoltaic* adalah piranti semikonduktor yang dapat merubah cahaya secara langsung menjadi arus listrik searah (DC) menggunakan kristal silikon (Si). Silikon tersebut dipasang pada posisi sejajar dalam sebuah panel kemudian tiap sambungan sel diberikan sambungan listrik.

Apabila sel silikon tersebut terpapar sinar matahari, maka pada sambungan tersebut akan mengalir arus listrik yang besarnya tergantung pada jumlah energi cahaya matahari yang mencapai silikon dan luas permukaan sel surya.

Keuntungan dari pembangkit listrik tenaga surya diantaranya [2].

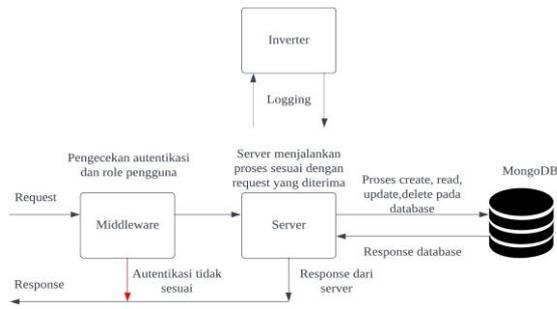
1. Energi yang digunakan dalam menghasilkan listrik merupakan energi yang tersedia dari alam dan secara cuma-cuma.
2. Perawatan *photovoltaic* mudah dan sederhana.
3. Tidak adanya perangkat yang bergerak dalam pembangkit listrik ini memberikan keuntungan kepada pengguna berupa tidak perlunya mengganti suku cadang dari pembangkit listrik ini.
4. Memberikan kenyamanan kepada pengguna karena dalam pengunannya tidak menghasilkan suara yang berisik dan berdampak negatif kepada lingkungan.
5. Dengan bantuan IoT, pembangkit ini dapat bekerja secara otomatis dan *monitoring* yang dilakukan dapat dilakukan dengan mudah.

C. *Dashboard*

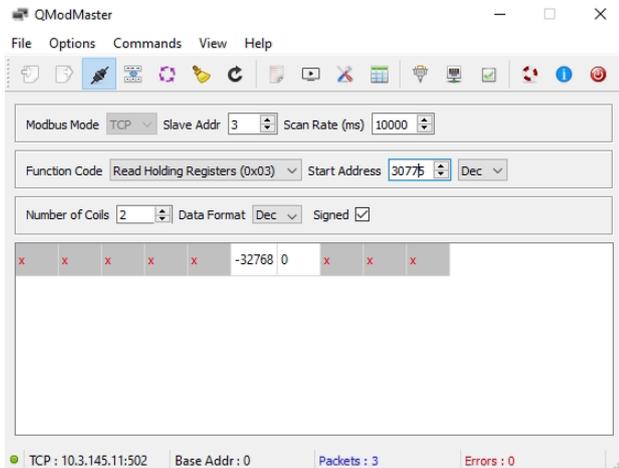
Dashboard menawarkan tampilan dari kumpulan data yang besar dan kompleks yang bisa dilihat hanya dengan sekilas saja. *Dashboard* menggabungkan representasi visual dan grafis untuk memberikan penyederhanaan terhadap data terkait sehingga pengamat *dashboard* mendapatkan gambaran umum tentang informasi yang paling penting atau relevan dengan waktu yang efisien. Hal ini membuat *dashboard* banyak digunakan pada banyak domain aplikasi seperti bisnis, keperawatan dan rumah sakit, kesehatan masyarakat, analisis pembelajaran, analisis perkotaan, analisis pribadi, energi, dan banyak lagi [3].

D. *JavaScript*

JavaScript dibuat pada Mei 1995 oleh Brendan Eich. Ide dari terbentuknya JavaScript adalah untuk membantu pembuatan web yang akan diimplementasikan ke dalam bahasa Java dan sebagai bahasa perekat untuk bagian bagian HTML sehingga menjadi lebih interaktif.



Gambar 2. Rancangan Kerja Server.



Gambar 3. Percobaan Pembacaan Address dengan QModMaster.

Dalam JavaScript, terdapat beberapa library yang dapat membantu pengembang web untuk membangun sebuah user interfaces. Dalam kasus ini, penulis akan menggunakan library yang bernama ReactJS. ReactJS adalah library JavaScript untuk membangun *user interfaces* yang dikelola oleh Facebook bersama dengan komunitas *developer* dan perusahaan. Penulisan komponen React menggunakan format penulisan “*JSX*” yang merupakan singkatan dari JavaScript XML. *JSX* memungkinkan *developer* untuk menuliskan HTML dalam JavaScript dan dapat langsung menyatukan HTML tersebut dengan perintah yang ada pada JavaScript. Hal ini memudahkan *developer* untuk memahami sintaks dari React menjadi lebih mudah [4].

E. Modbus TCP/IP

Modbus TCP/IP atau Modbus-TCP adalah protokol modbus RTU dengan TCP yang berjalan di ethernet. TCP/IP mengacu pada *Transmission Control Protocol* dan *Internet Protocol*, yang menyediakan media transmisi untuk layanan pesan modbus TCP/IP. TCP/IP memungkinkan blok data biner untuk dipertukarkan antara komputer. Fungsi utama TCP untuk memastikan bahwa semua paket data diterima dengan benar, sementara IP memastikan bahwa pesan-pesan ditujukan dan dialihkan dengan benar. Kombinasi TCP/IP hanya digunakan sebagai protokol *transport* dan tidak menentukan bagaimana data akan ditafsirkan karena hal itu merupakan tugas dari modbus. Singkatnya, modbus TCP/IP menggunakan TCP/IP dan ethernet untuk membawa data struktur pesan modbus antar perangkat yang kompatibel dengan menggabungkan jaringan fisik yaitu ethernet dan standar jaringan (TCP/IP) dan modbus sebagai metode merepresentasikan data.

Tabel 3. Perbandingan Volume Data.

Nama Inverter	Jumlah Data	Besar Data Sesudah Dikompres / Sebelum Dikompres
Sunny Tri Power	986687	61,31MB/308,13MB
Sunny Island 4.4M - 13	976461	29,95MB/199,48MB
Sunny Boy 2.0	986988	52,65MB/304,73MB

F. Server-Sent Events

Server-sent events merupakan kemampuan dari web untuk secara otomatis mendapatkan pembaruan dari server. Pada umumnya, halaman web harus mengirimkan permintaan (*request*) ke server untuk menerima data baru. Dengan *server-sent event*, server dapat mengirimkan data baru ke halaman web kapan saja.

Penggunaan *server-sent events* pada web dimampukan menggunakan perantara *EventSource* pada web. *EventSource* akan membuka koneksi ke server HTTP yang akan mengirimkan event yang berisikan data dalam format *text/event-stream*. Koneksi akan terus terbuka sampai dengan salah satu dari pengakses atau server memutuskan hubungan. Setelah koneksi dibuka, pesan dari server akan masuk ke web dalam bentuk events. *Server-sent events* hanya memampukan pengiriman pesan dalam satu arah yaitu dari server ke klien.

G. Scheduling

Scheduling merupakan proses memberikan akses ke sumber daya pada sistem. Kebutuhan akan *scheduling* muncul dari kebutuhan sistem komputer yang cepat seperti melakukan *multitasking* (melakukan lebih dari satu proses dalam waktu bersamaan) dan *multiplexing* (mengirimkan beberapa sinyal secara bersamaan). *Scheduling* adalah fungsi dasar pada *operating system* yang menentukan urutan proses yang akan berjalan saat terdapat beberapa proses yang perlu dijalankan. Tujuan dari *scheduling* antara lain [5].

1. Memaksimalkan *throughput* dengan mampu menjalankan jumlah maksimum proses per unit waktu.
2. Menghindari terjadinya pemblokiran terhadap suatu proses yang menyebabkan terjadinya *delay* dengan waktu yang tidak terbatas saat ataupun sebelum memproses tugas.
3. Meminimalkan pemborosan seperti waktu komputasi, memori dan *bandwidth* (*Overhead*).

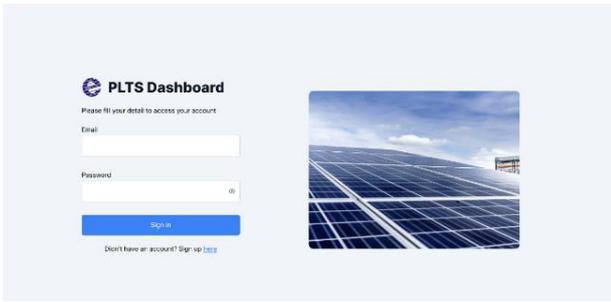
III. PERANCANGAN ALAT

A. Gambar dan Tabel

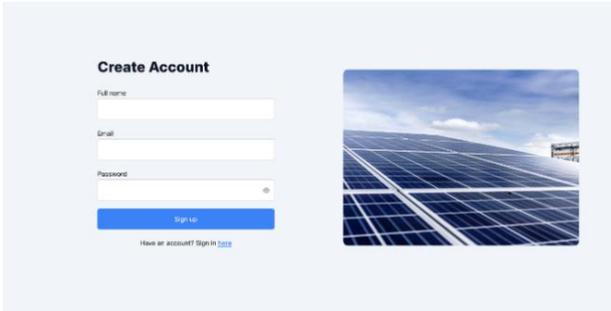
Pada sistem yang akan didesain, terdapat 2 sisi kebutuhan yang diperlukan antara lain kebutuhan perangkat keras seperti pada Tabel 1 dan perangkat lunak seperti pada Tabel 2.

B. Desain Sistem Monitoring

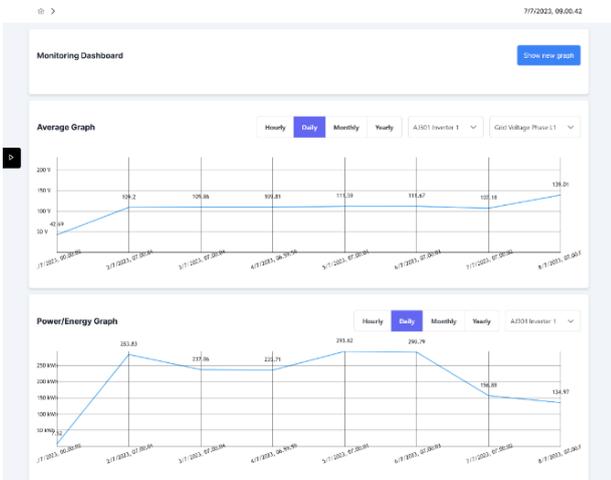
Sistem *dashboard monitoring* didesain untuk menggunakan Modbus TCP yang sudah terpasang pada inverter *photovoltaic* pada inverter Pembangkit Listrik Tenaga Surya (PLTS) seperti pada Gambar 1. Server akan membuat koneksi dengan modbus yang sudah terhubung dengan inverter *photovoltaic* dan mengirimkan permintaan data melalui kode fungsi modbus dan alamat register yang digunakan untuk menentukan data yang akan diterima. Modbus dan server terhubung menggunakan jaringan WAN yaitu WiFi ITS yang tersebar di kampus ITS. Permintaan



Gambar 4. Tampilan Halaman Login.



Gambar 5. Tampilan Halaman Daftar.



Gambar 6. Tampilan Halaman Utama Dashboard.

akan diteruskan ke modbus dan modbus akan melakukan permintaan data pada inverter *photovoltaic*. Inverter *photovoltaic* menerima permintaan data dari modbus dan mengirimkan respon modbus yang berisi data yang diminta dan mengirimkannya kembali ke modbus. Modbus akan menginterpretasikan data yang diterima dari inverter *photovoltaic* sesuai dengan dokumentasi pabrikan inverter dan akan diteruskan ke server. Server selanjutnya akan menerjemahkan data dari modbus sesuai dengan dokumentasi dari pabrikan inverter dan disimpan pada database.

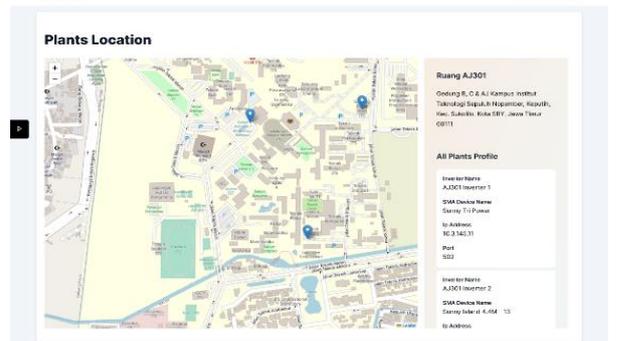
Untuk menampilkan data pada web, web akan melakukan permintaan data ke server melalui *endpoint* yang sudah disediakan pada server. Server akan menjalankan proses untuk menghasilkan data yang dibutuhkan seperti melakukan pembacaan terhadap data yang dibutuhkan dan akan dikirimkan kembali ke web untuk divisualisasikan.

C. Persiapan Database

Database yang digunakan adalah MongoDB dengan menggunakan localhost dan penyimpanan internal sebagai media akses dan menyimpan data. Data yang disimpan yaitu staff, profil inverter, posisi, dan data yang di-logging oleh

Inverter Name	SMA Device Name	IP Address	Port	Address
AJ201 Inverter 1	Sunny 10 Power	10.3.145.11	502	Ruang A3201 Gedung B, C & A1 Kampus Institut Teknologi Sepuluh Nopember, Kampus, Salsabila, Kota SEP, Jawa Timur 60111
AJ201 Inverter 2	Sunny Island 4.0W-12	10.3.145.14	502	Ruang A3201 Gedung B, C & A1 Kampus Institut Teknologi Sepuluh Nopember, Kampus, Salsabila, Kota SEP, Jawa Timur 60111
AJ201 Inverter 3	Sunny Day 2.0	10.3.145.15	502	Ruang A3201 Gedung B, C & A1 Kampus Institut Teknologi Sepuluh Nopember, Kampus, Salsabila, Kota SEP, Jawa Timur 60111

Gambar 7. Tampilan Daftar Inverter.



Gambar 8. Tampilan Halaman Map.

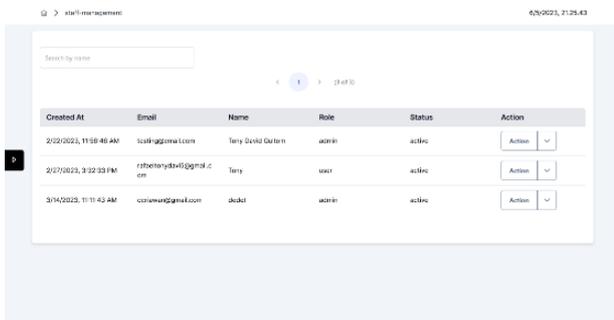
Name	Address	Latitude	Longitude
Gedung Rektorat ITS	Kampus ITS, Salsabila, 1 Rangs ITS, Hoeslye, Surabaya Kota SEP, Jawa Timur 60117	-7.28192523690525	112.7543071495644
Gedung Research Center	Kampus ITS, Salsabila, Kelle SEP, Jawa Timur 60117	-7.281925774442759	112.7547402509354
Pusat Robotika	Kampus ITS, Salsabila, Salsabila City, East Java 60117	-7.277872747354158	112.7541988472933
Ruang A3201	Gedung B, C & A1 Kampus Institut Teknologi Sepuluh Nopember, Kampus, Salsabila, Kota SEP, Jawa Timur 60111	-7.28134934810026	112.750444139672

Gambar 9. Tampilan Halaman Posisi

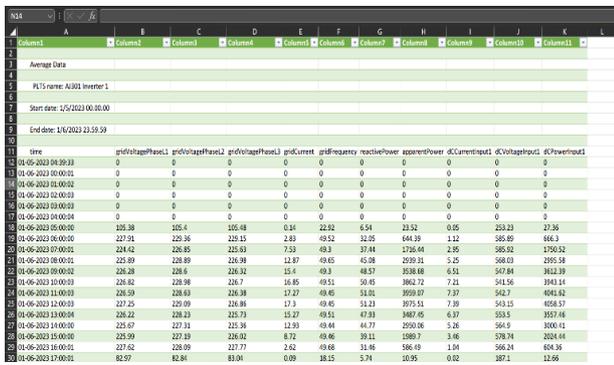
server setiap 5 detik. Dalam pembuatan database ini, komputer server perlu diinstall MongoDB Community Server sebagai *software* database dan juga MongoDB Compass sebagai *interface* untuk melihat sebaran data pada database. Saat melakukan instalasi dari MongoDB Community Server, terdapat beberapa hal yang perlu diketahui. Pengguna perlu mengatur lokasi folder dari “data directory” dan “log directory” dari database. Hal ini diperlukan dikarenakan MongoDB menggunakan kedua folder tersebut sebagai lokasi penyimpanan data.

D. Perancangan Server

Server berperan sebagai penghubung antara database dan web. Web akan melakukan permintaan data melalui *endpoint* API yang tersedia seperti pada gambar rancangan kerja yang ditunjukkan pada Gambar 2. Pada sisi server, permintaan tersebut akan diperiksa terlebih dahulu seperti kebenaran token dan kecocokan *role* pengguna. Apabila proses pengecekan berhasil server akan menjalankan proses sesuai dengan *endpoint* API yang diakses oleh pengguna termasuk melakukan akses terhadap database apabila dibutuhkan. Apabila terjadi kegagalan pada proses yang dijalankan, server



Gambar 10. Tampilan Halaman Manajemen Staff.



Gambar 11. File Hasil Pencatatan Data.

akan memberikan respon berupa kode *error* dan pesan *error*.

E. Percobaan Pembacaan Data dengan Qmodmaster

Pada penelitian, akan digunakan modbus sebagai media untuk melakukan *testing* pembacaan terhadap data yang dihasilkan dari inverter. Pada inverter, telah disediakan datasheet alamat modbus yang berisi deskripsi dan satuan yang dihasilkan dari data tersebut seperti pada percobaan yang ditunjukkan Gambar 3.

IV. ANALISA HASIL

A. Implementasi Server

Implementasi server merupakan tahap pembuatan server pada komputer sesuai dengan *schema* yang sudah didesain. Dengan menggunakan server *environment* NodeJS, pengguna menambahkan beberapa *library* untuk membantu menghubungkan web dengan server antara lain, mongoose untuk menghubungkan server dengan database MongoDB, modbus-serial untuk melakukan pembacaan data terhadap modbus, bcrypt untuk melakukan *hashing* terhadap password, express-jwt untuk melakukan enkripsi token untuk pengguna, ip untuk mendeteksi alamat ip, dan Typescript untuk meminimalisir *bug* pada kodingan dan melakukan *build* terhadap aplikasi server. Bahasa pemrograman yang dipakai adalah Typescript dan Javascript. Untuk melakukan *deploy* terhadap server yang sudah jadi, pengguna menggunakan *library* bernama PM2. PM2 merupakan sebuah *library* yang memungkinkan server untuk melakukan *restart* secara sendiri apabila terdapat *error* pada server. Selain itu, PM2 juga dapat melakukan *monitoring* terhadap aktivitas yang terjadi pada server dan juga mengoptimisasi server agar dapat sesuai dengan jumlah *thread* yang terdapat pada cpu komputer. oleh pengguna seperti pada Gambar 11.

Tabel 4.

Perbandingan Waktu Komputasi Variabel Rata-Rata Inverter.

Nama Inverter	Jumlah Data	Hourly	Daily	Monthly	Yearly
Sunny Tri Power	986687	156ms	703 ms	6,31s	6,11s
Sunny Island 4.4M - 13	976461	119ms	520 ms	4,56s	4,29s
Sunny Boy 2.0	986988	164ms	696 ms	6,29s	6,02s

Tabel 5.

Perbandingan Waktu Komputasi Data Power.

Nama Inverter	Jumlah Data	Hourly	Daily	Monthly	Yearly
Sunny Tri Power	986687	82ms	447ms	3,97s	3,96s
Sunny Island 4.4M - 13	976461	77ms	428ms	3,90s	3,89s
Sunny Boy 2.0	986988	78ms	449ms	4,01s	3,96s

Tabel 6.

Perbandingan Waktu Komputasi Data Energi Cluster.

Jumlah Data	Daily	Monthly	Yearly
2950136	3,19s	31,09s	31,14s

B. Volume Data

Tabel 3 menunjukkan volume data yang telah dikumpulkan oleh server terhitung pada tanggal 7 Maret 2023 – 16 Mei 2023.

C. Implementasi User Interface

Implementasi *user interface* merupakan tahap pembuatan antar muka yang sudah didesain. Pada web *monitoring* ini, *user interface* yang diimplementasikan antara lain:

1. Tampilan *Login* yang ditunjukkan oleh Gambar 1.
2. Tampilan Daftar yang ditunjukkan oleh Gambar 5.
3. Tampilan Halaman Utama *Dashboard* yang ditunjukkan oleh Gambar 6.
4. Tampilan Halaman Inverter yang ditunjukkan oleh Gambar 7.
5. Tampilan Halaman Map yang ditunjukkan oleh Gambar 8.
6. Tampilan Halaman Posisi yang ditunjukkan oleh Gambar 9.
7. Tampilan Halaman Manajemen Staff yang ditunjukkan oleh Gambar 10.

D. Implementasi Penyimpanan File Pencatatan

Implementasi penyimpanan file pencatatan memungkinkan pengguna untuk menyimpan data berupa file hasil rata-rata per jam dari pencatatan menggunakan Modbus TCP dengan rentang waktu tertentu. Data akan disimpan pada format *Comma Separated Value* (CSV). File tersebut dapat didownload pada halaman web *monitoring* dan dapat ditentukan rentang waktu data yang dibutuhkan

E. Pengujian Waktu Komputasi

Pengujian ini bertujuan untuk mengukur lama waktu komputasi dari server untuk menghasilkan sebuah data

seperti yang ditunjukkan Tabel 1, Tabel 5, dan Tabel 6. *Hourly* merupakan perbandingan per jam. *Daily* merupakan perbandingan per hari. *Monthly* merupakan perbandingan per bulan. *Yearly* merupakan perbandingan per tahun.

V. KESIMPULAN

Pada penelitian ini telah didesain sebuah web *monitoring* yang berfungsi untuk memproses dan melakukan pencatatan terhadap data yang dihasilkan pada inverter pada PLTS di kawasan ITS. Pembacaan data dilakukan melalui server yang melakukan permintaan data ke modbus setiap 5 detik dan disimpan pada *database*. Data yang disimpan akan ditampilkan pada web *monitoring* berupa data yang sudah diproses ataupun data *real-time*.

Setelah dilakukan pembuatan web dan juga server dan melalui tahapan pengujian, web dan server sudah dapat digunakan dan dapat digunakan pada daerah yang sudah terhubung dengan jaringan internet ITS. Penelitian ini menghasilkan beberapa kesimpulan antara lain: (1) Pada penelitian ini telah dirancang server dan web mampu untuk melakukan pembacaan terhadap inverter panel surya yang sudah terpasang dan dapat melakukan visualisasi dan pemrosesan data terhadap data tersebut; (2) Semakin besar

range data yang dipilih pada grafik, maka akan menghasilkan waktu tunggu yang lebih lama juga. Hal ini terjadi karena banyaknya data yang perlu dibaca oleh sebuah *database*; (3) Penggunaan *index* yang tepat pada *database* dapat membantu pemrosesan data pada *database*. *Index* dapat mengerucutkan data yang disimpan pada *database* sehingga *database* tidak perlu melakukan pembacaan terhadap semua data yang tersedia; (4) Penggunaan *query* pada *database* dapat memengaruhi kecepatan pemrosesan data pada *database* sehingga diperlukan keefisienan pada *query*.

DAFTAR PUSTAKA

- [1] J. Salazar Soler and S. Silvestre Bergés, *Internet of Things*. Czech Republic: European Virtual Learning Platform for Electrical and Information Engineering, 2017, ISBN: 978-80-01-06232-6.
- [2] R. Salman, "Analisis perencanaan penggunaan sistem Pembangkit Listrik Tenaga Surya (PLTS) untuk perumahan (solar home system)," *Maj. Ilm. Bina Tek.*, vol. 1, no. 1, pp. 46–51, 2013.
- [3] B. Bach *et al.*, "Dashboard Design Patterns," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 1, pp. 342–352, 2022.
- [4] S. Bin Uzayr, N. Cloud, and T. Ambler, *JavaScript Frameworks for Modern Web Development*. New York: Springer, 2019.
- [5] M. Ramakrishna and G. P. R. Rao, "Efficient round robin CPU scheduling algorithm for operating systems," *Int. J. Innov. Technol. Res.*, vol. 1, pp. 103–109, 2013.