

Terbit online pada laman web jurnal : <http://metal.ft.unand.ac.id>**METAL: Jurnal Sistem Mekanik dan Termal**

| ISSN (Print) 2598-1137 | ISSN (Online) 2597-4483 |



Artikel Penelitian

Rancang Bangun Sistem Kontrol dan Monitoring Data Turbin Angin Berbasis Website Menggunakan Raspberry Pi 3B+

Meiki Eru Putra ^a, Zulkifli Amin ^b, Islahuddin ^a, Sanny Ardhy ^a^a Program Studi Teknik Mesin, Universitas Dharma Andalas, Simpang Haru, Kota Padang 2500, Indonesia^b Jurusan Teknik Mesin, Universitas Andalas, Kampus Limau Manis, Kota Padang 25163, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 05 September 2020

Revisi Akhir: 15 September 2020

Diterbitkan Online: 08 Oktober 2020

KATA KUNCI

Sitem kontrol

Monitoring

Turbin Angin

Website

Raspberry Pi 3B+

KORESPONDENSI

E-mail: meikieruputra@gmail.com

A B S T R A C T

Indonesia has considerable wind energy potential. Energy generation system performance can be monitored and controlled through the Internet of Things (IoT) technology. However, research and development of IoT technology in wind turbines in Indonesia is still low. Therefore, in this study, the author seeks to conduct research related to the development of a website-based wind turbine control and data monitoring system. This research uses a 3 blade horizontal axis wind turbine (NACA 2410). The monitoring system uses the Raspberry Pi 3B + microcontroller and several sensor devices such as the IR LM393 speed sensor to measure wind speed and shaft rotation speed, and the INA219 current-voltage sensor to measure the voltage and current of a 350 Watt DC generator. The website was created using the Laravel 5.8 PHP framework and Chart.js. Furthermore, the sensor calibration process and system testing are carried out. Based on the test results, it was found that the monitoring data system was functioning properly. The data has been successfully sent to the server and can be monitored in real-time via the website. Based on the results of the IR LM393 sensor calibration test, the linear regression equation $y = 0.7881x + 13.837$, and the value of $R^2 = 0.9934$ is obtained. And the value of $R^2 = 0.9934$ indicates that the sensor calibration process is well correlated. The results of testing the control system and monitoring of wind turbine data monitored through the website for 1 x24 show that the highest wind potential occurs at 13.00 to 14.00 WIB, namely at a wind speed of 3.74 m / s.

1. PENDAHULUAN

Potensi energi angin di Indonesia sangat besar. Akan tetapi, hingga saat ini pemanfaatan Energi Baru Terbarukan (EBT) ini masih rendah. Revolusi Industri 4.0 menuntut perubahan besar dalam pemanfaatan internet sebagai media pendukung dalam penyelesaian berbagai permasalahan yang ada pada saat ini. Setiap perangkat, sensor atau sistem pembangkit energi dapat terhubung melalui internet dengan memanfaatkan teknologi *Internet of*

Things (IoT). Sehingga setiap data yang diperoleh melalui sensor, performa alat, kondisi terkini dan analisa data serta permasalahan yang terjadi pada sebuah sistem pembangkit dapat diketahui secara langsung (*live*) melalui perangkat komputer yang terhubung melalui internet. Namun, pemanfaatan teknologi *IoT* pada pembangkit listrik EBT khususnya untuk Pembangkit Listrik Tenaga Bayu (PLTB) masih sangat minim.

Pada tahun 2016, Muhamamad Hanif Abdillah, dkk telah melakukan penelitian dan pemanfaatan

teknologi *IoT* untuk memonitor secara *real-time* penyimpanan energi listrik dari *wind turbine* Lentera Angin Nusantara (LAN) [1]. Perangkat monitoring yang digunakan berbasis komunikasi *machine to machine* (M2M) dan *Internet of Things* (*IoT*) dengan menggunakan protokol *MQTT* (*Message Queueing Telemetry Transport*). *MQTT* Client Publisher adalah mikrokontroler Arduino Uno yang tertanam program C++ atau platform perangkat lunak berbasis mikrokontroler dengan menggunakan *library PubSubClient*. Data arus dan tegangan diperoleh dari sensor arus dan tegangan. Kemudian diakuisisi oleh mikrokontroler, setelah itu di-*publish* secara *real-time* ke *broker* dengan nama topik tertentu “sensors/energymonitor”. Sebagai *MQTT broker* adalah *HiveMQ*. Pada *broker* data energi listrik pada topik tertentu yang dikirimkan oleh mikrokontroler dan di simpan dalam *log file*. Namun, penelitian tersebut masih terbatas pada monitoring data arus dan tegangan penyimpanan dengan menggunakan protokol *MQTT*.

Ada juga yang telah memanfaatkan teknologi *smart phone android* untuk memonitor turbin angin [2]. Monitoring turbin angin yang dilakukan mencakup parameter kecepatan angin dan kecepatan putaran turbin. Data parameter ini diperoleh dari sensor kecepatan angin dan kecepatan putaran yang diolah dengan *arduino mega 2560*. Data dari *arduino* ini dikirim melalui modul *Bluetooth HC-04* untuk ditampilkan pada *smartphone android*. Hasil pengujian menunjukkan bahwa sistem monitoring turbin angin yang diusulkan telah bekerja dengan baik. Data kecepatan angin dan kecepatan putaran turbin yang ditampilkan pada *android* sudah persis sama dengan data pada alat ukur. Namun, penelitian tersebut terbatas pada menggunakan teknologi *bluetooth* yang hanya mampu di akses dengan perangkat *android* pada jarak yang terbatas.

Selain itu, ada beberapa penelitian yang memanfaatkan *IoT* berbasis web untuk proses monitoring turbin angin. Seperti yang dilakukan oleh Handarly. D dan Jefri Lianda, pada penelitiannya yang berjudul “Sistem Monitoring

Daya Listrik Berbasis *IoT (Internet of Thing)*” [3]. Penelitian tersebut bertujuan untuk mendapatkan informasi-informasi yang berhubungan dengan pengukuran energi listrik antara lain daya semu (VA), tegangan (V), dan Arus (A) secara *real-time* yang dapat diakses dari Jaringan Internet. Untuk menghubungkan ke internet alat ini menggunakan *ethernet shield*, dan untuk tampilan monitoring di internet menggunakan *Ubidot*. Sistem monitoring ini dalam 1 menit menghasilkan data sebanyak 60 data, data dimonitoring dalam waktu per detik. Untuk nilai perbandingan antara daya yang terbaca pada tampilan monitoring dengan alat ukur watt meter memiliki tingkat akurasi diatas 90 % dengan persentase error 2,96 –7,28 % [4]. Meskipun *platform Ubidot* tersebut bersifat gratis, namun fiturnya banyak yang telah dibatasi, untuk mendapatkan fitur penuh maka pengguna harus berlangganan aplikasi tersebut.

Winasis. Dkk, mencoba menyajikan purwarupa sistem monitoring nirkabel baterai dan beban pada PLTH surya – angin berbasis website yang mampu memantau secara remote kinerja PLTH [4]. Rancangan perangkat keras sistem monitoring yang digunakan meliputi: perangkat wireless node pemantauan baterai, perangkat wireless node pemantauan beban dan perangkat pusat pemantauan. Besaran energi listrik pada baterai dan beban meliputi: tegangan, arus, daya, frekuensi, dan faktor daya diukur dan diproses pada node pemantauan untuk dikirimkan ke pusat pemantauan secara nirkabel melalui wireless transceiver menggunakan protokol komunikasi 802.11 b/g/n. Pengiriman data hasil pemantauan ke jaringan internet dilakukan oleh pusat pemantauan melalui sebuah modem GSM. Rancangan prototipe sistem monitoring yang diusulkan diimplementasikan pada skala laboratorium dengan menggunakan DC power supply (*battery charger*) sebagai sumber arus DC, baterai nominal 2x12V dan inverter 1 kW. Hasil pengujian menunjukkan bahwa keseluruhan perangkat node monitor dan sentral monitor dapat bekerja dengan baik. Fungsi pencatatan (*logging*), pengolahan data, serta penampilan dan akses data secara lokal maupun melalui jaringan internet pada penelitian ini dikembangkan dengan menggunakan

layanan *Thingspeak* yang merupakan open data platform untuk IoT. Aplikasi yang dapat dilakukan antara lain: mengumpulkan data sensor secara real time dan mengirimkan ke cloud, penyimpanan data hasil pengukuran, analisis dan visualisasi data, fungsi alarm, dan penjadwalan. Berdasarkan informasi yang diperoleh dari website resmi *Thingspeak*, layanan untuk versi gratis *Thingspeak* hanya tersedia untuk proyek kecil non-komersial (~ 8.200 pesan/hari) dan tidak diizinkan untuk skalabel lebih besar.

Beberapa penelitian lain juga telah membahas berbagai aplikasi sistem monitoring data pembangkit energi terbarukan. Shariff, dkk melakukan penelitian terkait dengan pemantauan sistem *PV* di area *remote* menggunakan modem *GSM* sebagai transmisi data [5]. Selain itu, pemanfaatan teknologi website untuk monitoring data sistem pembangkit listrik terbarukan juga dilakukan oleh Li [6] dan Kopacz [7]. Soetedjo [8] meneliti implementasi web – scada untuk memonitor dan mengontrol sistem pembangkit hibrida surya – angin secara *remote* melalui jaringan internet. Pada penelitian ini antara sensor, *remote terminal* dan komputer server masih memakai jaringan kabel menggunakan komunikasi serial dan local area network. Beberapa penelitian telah membahas berbagai aplikasi sistem monitoring pembangkit yang memanfaatkan sumber energi terbarukan. Namun pada penelitian tersebut masih memakai jaringan kabel menggunakan komunikasi serial dan local area network untuk mentransmisikan data.

Oleh sebab itu, pada penelitian ini penulis berupaya melakukan penelitian terkait pengembangan perangkat lunak *IoT* berbasis website yang bersifat *open-source* untuk monitoring data turbin angin. Penelitian ini dimulai dari merancang sistem kontrol dan website monitoring data turbin angin menggunakan *Raspberry Pi 3B+* sebagai mikrokontroler. Mikrokontroler bertugas untuk membaca serta mengolah data dari sensor dan mengirimkannya ke *database server* sehingga dapat diakses dimana saja. Data yang telah dikirim akan ditampilkan secara *real-time* berupa tabel dan grafik analisis

pada website. Sensor yang digunakan antara lain adalah *IR LM319 Speed Sensor* dan *INA219 Current Voltage Sensor*.

Penelitian terkait penggunaan *Raspberry Pi* sebagai mikrokontroler dalam monitoring data telah dilakukan oleh Muck dan Homan [9]. Dimana stasiun cuaca dirakit menggunakan *SparkFun Weather Shield* dan *Weather Meter* dan *Arduino Uno R3* untuk mengumpulkan parameter cuaca. Data yang dikumpulkan dari sensor kemudian disimpan ke dalam *Google Cloud SQL* menggunakan *Raspberry Pi 3 Model B* yang bertindak sebagai gateway di antara sensor tersebut dan dilakukan analisis data cuaca.

Website monitoring data turbin angin ini akan dibuat dengan menggunakan teknologi *framework Laravel* yang bersifat *open-source* dan mudah untuk dikembangkan kedepan dalam skala yang jauh lebih besar. *Laravel* adalah *framework PHP* yang paling canggih, elegan dan membuat proses pengembangan menjadi sederhana. *Framework Laravel* memiliki tiga komponen arsitektur yang saling berhubungan yang disebut *Model, View* dan *Controller*. *Model* terdiri dari data dan logika bisnis. Kelas *controller* digunakan untuk mengatur perilaku logika penanganan permintaan. Paket pihak ketiga dapat diintegrasikan dengan mudah ke dalam aplikasi *Laravel* dengan menulis beberapa perintah *artisan*. Paket ini menghemat banyak waktu pengembangan. Sebagian besar paket ini telah diuji dengan baik, stabil, dan tersedia secara gratis [10].

Dalam pengembangan *IoT*, pengolahan dan analisa data yang terfokus pada pengukuran data yang besar membutuhkan sensor yang banyak dan dapat melakukan pengukuran secara *real-time* merupakan hal yang menantang [11]. Stonebraker dkk. mengatakan bahwa pada sebuah alat/mesin pemrosesan secara *real time* harus memenuhi persyaratan yaitu [11] :

- a. Kemudahan dalam akses data, yang mengacu pada pemrosesan data saat bepergian tanpa perlu biaya penyimpanan data.
- b. Menangani berkas/data yang tidak teratur, hilang, dan tertunda.

- c. Dapat diakses secara berulang dengan hasil yang tetap setelah memproses serangkaian atau sejumlah berkas data.
- d. Selalu terhubung dan data tetap tersimpan yang terintegrasi dengan sistem database yang telah tertanam.
- e. Jaminan ketersediaan data yang tinggi, dengan sistem keamanan *real-time* dan mekanisme pencadangan panas.
- f. Mendukung penskalaan otomatis dan dapat dioperasikan dengan aplikasi (*software*).

2. METODOLOGI

Turbin angin yang digunakan pada penelitian ini adalah turbin angin tipe horizontal 3 sudu dengan profil *Airfoil NACA 2410* (Gambar 1). Sistem transmisi turbin ini terdiri dari *hub*, *bearing*, poros yang dihubungkan ke *pulley*. *Pulley* utama dihubungkan dengan *Belt* sehingga *pulley* pada poros memutar generator. Generator yang digunakan adalah generoator *DC 350 Watt*. Untuk mengukur performa turbin angin secara *real-time* dibutuhkan sistem kontrol dan perangkat lunak berbasis web yang dapat memantau data tersebut melalui berbagai perangkat komputer atau *mobile*.

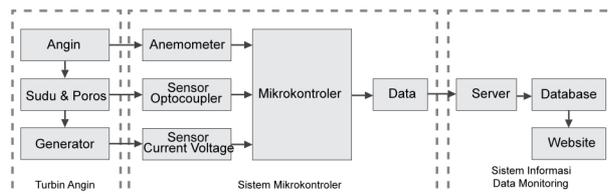


Gambar 1. Turbin Angin Tipe Horizontal 3 Sudu

2.1. Perancangan Sistem

Pada tahapan ini dilakukan perancangan diagram blok sistem, yaitu: perancangan sistem kontrol dan sistem informasi data monitoring berbasis web. Sistem ini dibagi kedalam 3 komponen utama, yaitu: turbin angin, mikrokontroler, dan sistem informasi data monitoring turbin angin berbasis

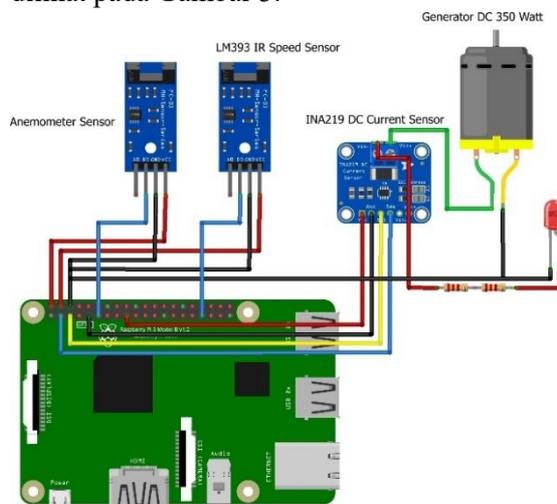
website (Gambar 2). Turbin angin berfungsi sebagai sumber data yang akan diolah olah sensor dan dikirim ke server untuk ditampilkan pada halaman website yang dapat diakses secara *real-time* melalui perangkat komputer, laptop, atau *hand phone*. Data yang dikukur dan dimonitoring antara lain adalah kecepatan angin, kecepatan putaran poros turbin, voltase dan arus dari generator.



Gambar 2. Diagram Blok Sistem

2.1.1. Perancangan Sistem Kontrol

Sistem kontrol turbin angin ini menggunakan Raspberry Pi 3B+ sebagai mikrokontroler, sensor anemometer sebagai sumber data kecepatan angin, sensor *optocoupler* untuk mengukur kecepatan putaran poros turbin, dan sensor *current voltage INA-219* untuk mengukur arus dan voltase yang dihasilkan generator. Sinyal yang yang diperoleh dari sensor dikirim ke mikrokontroler. Mikrokontroler akan mengolah data sinyal yang diperoleh dari sensor menggunakan bahasa pemrograman Python untuk dikirim ke web server. Rancangan *wiring diagram* sistem kontrol dapat dilihat pada Gambar 3.



Gambar 3. *Wiring Diagram* Sistem Kontrol Data Monitoring Turbin Angin

Anemometer menggunakan *Infrared (IR) LM393 Speed Sensor* untuk membaca kecepatan angin pada Turbin. Pin digital (D0) pada Sensor tersebut dihubungkan ke pin GPIO17 pada Raspberry Pi 3B+. Untuk membaca kecepatan poros yang dihasilkan oleh bilah turbin juga menggunakan sensor *IR LM319 Speed Sensor*. Piringan yang memiliki celah sebanyak 18 buah di pasang pada poros sehingga ketika poros berputar sensor tersebut dapat membaca jumlah celah terang dan gelap dalam setiap satu kali putaran. Pin digital (D0) pada sensor tersebut dihubungkan ke pin GPIO 13 pada Raspberry Pi 3B+.

Untuk mengukur voltase dan arus yang dihasilkan generator menggunakan *INA219 DC Current Sensor*. Sensor tersebut menggunakan sistem I2C yang terdiri dari saluran *SCL (Serial Clock)* dan *SDA (Serial Data)* yang membawa informasi data antara I2C dengan pengontrolnya. Pin *SCL* pada sensor INA219 dihubungkan ke pin *GPIO3 (SCL1)* yang tersedia pada Raspberry Pi 3B+. Sedangkan pin *SDA* pada sensor INA219 dihubungkan dengan pin *GPIO2 (SDA1)* yang terdapat pada Raspberry Pi 3B+.

2.1.2. Perancangan Website Monitoring Turbin Angin

Proses perancangan dan pembuatan aplikasi website data monitoring turbin angin terdiri dari beberapa tahapan yaitu:

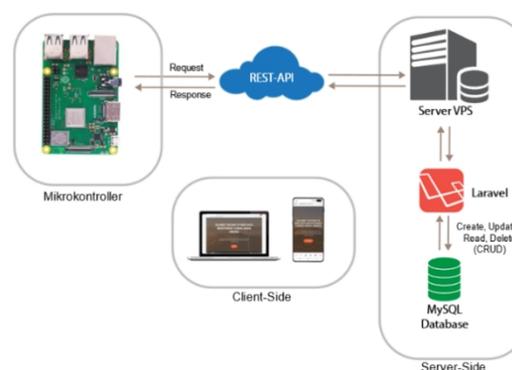
1. Pemilihan teknologi website dan perancangan sistem

Pada tahapan ini dilakukan pemilihan alternatif teknologi website yang akan digunakan. Teknologi website yang akan digunakan bersifat *open-source*, ringan, aman, mudah dalam pembuatan, mudah dalam pengembangan ke skala yang lebih besar. Selain itu, teknologi yang digunakan diharapkan juga memiliki dukungan dokumentasi yang baik dan fitur-fitur pendukung lainnya yang dibutuhkan dalam aplikasi data monitoring. Berdasarkan pertimbangan tersebut, penulis menetapkan pilihan pada teknologi website berbasis

bahasa pemrograman *PHP* menggunakan *framework* Laravel 5.8.

Framework Laravel adalah salah satu *framework PHP* yang cukup populer saat ini, bersifat *open-source*, ringan, tingkat keamanan yang baik, mudah dalam pembuatan dan pengembangan ke dalam skala yang lebih besar dan kompleks, serta didukung dengan fitur-fitur dan dokumentasi yang cukup lengkap untuk mengembangkan aplikasi data monitoring turbin angin berbasis website.

Untuk menampilkan data grafik secara *real-time* digunakan *JavaScript library* yaitu *Chart.js*. *Chart.js* adalah pustaka *JavaScript open-source* gratis untuk visualisasi data, yang mendukung 8 jenis *chart* yaitu: *bar, line, area, pie, bubble, radar, polar, and scatter*. Skema diagram sistem informasi monitoring data turbin angin berbasis website ini dapat dilihat pada Gambar 4.



Gambar 4. Skema Diagram Website Monitoring Data Turbin Angin

2. Perancangan arsitektur *database* website

Untuk menyimpan data yang dikirim oleh mikrokontroler ke server digunakan database *MySQL* untuk ditampilkan kembali dalam bentuk tabel dan grafik oleh *framework Laravel* dan *Chart.js*. Rancangan arsitektur database menggunakan *MySQL Workbench*.

Database terdiri dari 4 tabel (Gambar 5), yaitu tabel data untuk menyimpan data informasi dari turbin atau mikrokontroler, tabel *user* untuk menyimpan informasi pengguna dan *password*, tabel *password_reset* untuk

menyimpan data terkait reset *password* oleh pengguna, dan tabel *migrations* untuk proses migrasi data dari fitur migrasi yang terdapat pada *framework* Laravel.



Gambar 5. Aristektur Database Wesite Monitoring Data Turbin Angin

3. Perancangan *user interface* halaman website.

User interface (UI) atau antarmuka pengguna website dirancang dan dibuat menggunakan template Bootstrap yang berisi kode *HTML* dan *CSS* yang bersifat *open-source*, gratis, dan dapat di kostomisasi. Halaman website terdiri dari dari halaman *welcome* dan *dashboard*. Halaman *welcome* merupakan halaman awal yang pertama kali muncul ketika website monitoring data turbin angin diakses. Halaman ini menyediakan informasi terkait website yang diakses dan tombol untuk menuju ke halaman *dashboard* (Gambar 6).

Halaman *dashboard* adalah halaman yang menyediakan data tubin angin secara *real-time*. Halaman dashboard ini terdiri dari beberapa bagian yaitu:

1. Rangkuman data rata-rata turbin angin yang terdiri dari kecepatan angin rata-rata, voltase rata-rata, kecepatan putaran poros rata-rata, dan daya rata-rata (Gambar 7).
2. Grafik data *real-time* turbin angin. Pada grafik ini terdapat menu untuk memilih jenis grafik berdasarkan dan jenis variable yang akan ditampilkan seperti voltase, arus, daya, kecepatan angin, dan kecepatan putaran poros. Selain itu, grafik juga dapat ditampilkan berdasarkan waktu

pada seiap varibel data (60 menit terakhir, 24 jam terakhir, 7 hari terakhir terakhir, 30 hari terakhir, 12 bulan terakhir, dan semua data) (Gambar 8).

3. Tabel data turbin angin. Pada bagian ini ditampilkan semua varibel data turbin angin yang dikirim oleh mikrokontroller ke server dan diurutkan berdasarkan waktu terakhir data diambil (Gambar 9).



Gambar 6. Tampilan UI Halaman *Welcome*.



Gambar 7. Tampilan UI Halaman *Dashboard* – Data Rata-rata Turbin Angin.



Gambar 8. Tampilan UI Halaman *Dashboard* – Grafik Data *Real-time* Turbin Angin.

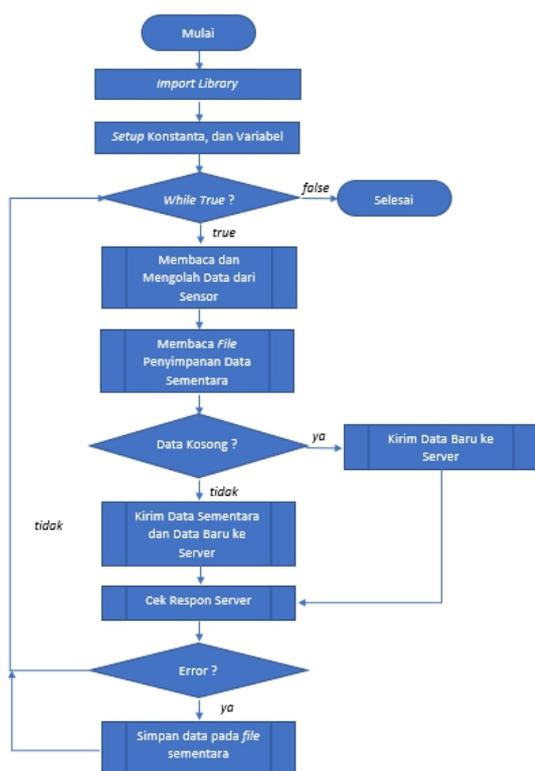
Waktu	Kecepatan Angin	Voltage	Arus	Daya	Putaran Poros	Efisiensi
13 Jan 2020 14:01:35	2,53	45,54	0,69	31,42	0	72,99 %
13 Jan 2020 14:01:34	2,53	45,63	0,69	31,55	1	72,98 %
13 Jan 2020 14:01:32	2,86	45,54	0,69	31,42	0,87	84,6 %
13 Jan 2020 14:01:32	2,52	45,54	0,69	31,42	5,33	72,35 %
13 Jan 2020 14:01:30	2,23	45,54	0,69	31,42	45	50,22 %
13 Jan 2020 14:01:29	2,87	45,54	0,69	31,42	26	85,51 %
13 Jan 2020 14:01:29	2,17	45,83	0,69	31,55	35,33	49,92 %
13 Jan 2020 14:01:28	2,58	45,63	0,69	31,55	37,67	76,74 %
13 Jan 2020 14:01:28	2,43	45,63	0,69	31,55	32,33	84,54 %
13 Jan 2020 13:01:55	1,34	34,08	0,52	17,57	40,67	19,41 %
13 Jan 2020 13:01:56	1,76	33,99	0,52	17,5	39	44,24 %
13 Jan 2020 13:01:54	1,71	33,99	0,52	17,5	29,67	40,6 %
13 Jan 2020 13:01:54	1,75	33,99	0,52	17,5	30,33	43,27 %

Gambar 9. Tampilan UI Halaman Dashboard – Grafik Data Real-time Turbin Angin.

2.2. Pembuatan Program

2.2.1. Pembuatan Program Sistem Kontrol pada Raspberry Pi

Program dirancang agar dapat membaca dan mengolah data dari sensor dan disimpan kedalam format *json* untuk dikirim ke server. Diagram alir program dapat dilihat pada 10.



Gambar 10. Diagram Alir Program Data Monitoring Turbin Angin Pada Raspberry Pi

Pada tahapan awal dilakukan proses import *library* pendukung yang dibutuhkan yang tidak tersedia pada *library* standar Python. Untuk pengiriman

data dari Raspberry Pi ke *API server* menggunakan *Request Library*. Sedangkan untuk membaca sinyal yang dikirim oleh sensor *INA219* menggunakan *library Adafruit INA219*.

Setelah proses impor *library* yang dibutuhkan, selanjutnya adalah mendefinisikan konstanta dan variabel yang akan digunakan. Setelah itu dilakukan proses perulangan tanpa henti (*infinite loop*) untuk membaca dan mengolah data dari sensor serta mengirimnya ke server secara *real-time* dan terus menerus. Jika terdapat data sementara yang tersimpan sebelumnya pada file penyimpanan sementara, maka sistem akan mengirim data sementara tersebut serta data terbaru ke server. Jika tidak terdapat data sementara yang tersimpan maka program akan mengirim data baru yang diperoleh dari sensor ke server. Selain itu, program juga akan membaca dan menerima respon dari server setiap proses pengiriman data. Jika server mengirimkan respons berupa *error code* maka data yang tidak berhasil terkirim tersebut akan disimpan pada file penyimpanan sementara dan membaca kembali sinyal baru dari sensor untuk diolah dan dikirim kembali ke server. Jika tidak ada respon *error* yang berasal dari server maka program akan membaca kembali sinyal yang diperoleh dari sensor untuk diolah dan dikirim kembali ke server.

2.2.2. Pembuatan Website Menggunakan Framework Laravel

Website dibuat menggunakan *framework* Laravel 5.8 yang berbasis *PHP*. Untuk menampilkan grafik *real-time* data turbin angin menggunakan *library* JavaScript yaitu *Chart.js*. *Framework* Laravel mengadopsi konsep arsitektur *MVC (Models, Views, Controllers)* dalam pembuatan sebuah website untuk mempermudah dalam pengembangan dan *scale-up* sistem. Oleh karena itu, dalam pembuatan website monitoring data turbin angin ini juga menggunakan konsep tersebut.

Komponen *models* merupakan komponen *MVC* yang berfungsi untuk mengatur setiap program yang akan mengakses database yang akan digunakan oleh *controller* dan *view* untuk menampilkan Data. Komponen *controllers* adalah

komponen *MVC* yang berfungsi untuk mengontrol akses terhadap data pada model untuk ditampilkan pada *views*. Komponen *views* berfungsi untuk menampilkan data yang diperoleh dari *controllers* kedalam tampilan halaman *HTML (Hypertext Markup Language)*.

Dalam pembuatan website ini dibagi kedalam beberapa tahapan, yaitu:

1. Pembuatan *database* monitoring data turbin angin menggunakan aplikasi *phpMyAdmin* pada server.
2. Instalasi *framework Laravel 5.8* pada server.
3. Konfigurasi koneksi database dari *Laravel* ke database *MySQL*.
4. Proses migrasi database dari *Laravel* ke database *MySQL*.
5. Instalasi *library Chart.js* pada *Laravel 5.8*.
6. Pembuatan *script* program pada *route* untuk mengatur akses halaman website.
7. Pembuatan *script* program pada komponen *controllers*.
8. Pembuatan *script* program pada komponen *models*.
9. Pembuatan *script* program pada komponen *views*.

Program lengkap website monitoring data turbin angin dapat di akses melalui link github berikut ini: https://github.com/meiki/data_monitoring_pltb_unidha/website.zip.

2.3. Kalibrasi Sensor

Sebelum dilakukan proses pengujian, dilakukan terlebih dahulu proses kalibrasi terhadap sensor-sensor yang digunakan. Proses kalibrasi dilakukan pada sensor *IR LM393*. Sensor *IR LM393* membaca banyaknya jumlah cahaya yang lewat pada sebuah piringan (*disc encoder*) sebagai sinyal yang akan dikirimkan ke mikrokontroler. *Disc encoder* terbuat dari bahan plastik yang dibentuk melingkar dengan jumlah celah yaitu 18 (Gambar 11).



Gambar 11. *Disc Encoder*

Kalibrasi pada sensor *IR LM393* dilakukan dengan memasang *disc encoder* pada sebuah poros dan poros tersebut diputar dengan menggunakan mesin bubut pada kecepatan putaran konstan. Sensor *IR LM393* diletakan pada posisi yang dekat dengan celah pada *disc encoder* untuk menangkap jumlah celah yang terbaca saat *disc encoder* berputar. Jumlah celah yang terbaca oleh sensor akan dikonversi kedalam satuan rotasi per menit (*rpm*) dengan menggunakan persamaan berikut:

$$NS = \frac{\left(\frac{Na}{Nk}\right) \times 60}{Pr} \quad (1)$$

Kecepatan putaran poros atau *disc encoder* dari hasil perhitungan jumlah celah yang dibaca oleh sensor (*Na*) pada periode yang ditetapkan (*Pr*) dibagi dengan jumlah celah yang terdapat pada *disc encoder* dalam waktu 1 menit (60 detik), sehingga diperoleh kecepatan putaran dalam satuan *rpm*. Nilai rata-rata yang terbaca oleh sensor akan dibandingkan dengan nilai kecepatan putaran pada mesin bubut. Kemudian dilakukan proses kalibrasi dengan menambahkan *offset value* pada *script* program. Pengujian dilakukan secara berulang pada variasi kecepatan putaran 45 *rpm* hingga 255 *rpm*. Proses kalibrasi sensor *IR LM393* dapat dilihat pada Gambar 12.



Gambar 12. Proses Kalibrasi *IR LM393 Speed Sensor*

2.4. Pengujian

Pengujian dilakukan dengan memasang perangkat turbin dan mikrokontroller di lantai 5 gedung C Universitas Dharma Andalas yang berlokasi di Jl. Sawahan No.103A Simpang Haru, Kota Padang. Pengujian dilakukan selama 1 x 24 jam. Raspberry Pi dihubungkan dengan koneksi internet melalui perantara modem GSM. Pengujian dilakukan untuk mengetahui apakah sensor bekerja dengan baik dan data dapat dikirim dan ditampilkan secara *real-time* pada website. Pengujian ini menggunakan metoda *White Box Testing* dan *Black Box Testing*. Pengujian menggunakan metoda *White Box Testing* dilakukan selama proses pembuatan blok kode aplikasi. Dan metode *Black Box Testing* dilakukan dengan melihat apakah semua fitur pada website monitoring data turbin angin berjalan dengan baik dan data dapat ditampilkan dengan baik di sisi pengguna.

3. HASIL DAN PEMBAHASAN

3.1. Program Sistem Kontrol Monitoring Data Turbin Angin pada Raspberry Pi

Program dibuat menggunakan bahasa pemrograman *Python* dan aplikasi *Thonny* yang tersedia pada Raspberry Pi 3B+. Program telah dibuat sesuai dengan diagram alir program seperti yang telah dijelaskan pada sub-bab 2.2.1 dan telah dilakukan uji coba. Program lengkap telah di unggah dan dapat diakses di pada link github berikut ini:

https://github.com/meiki/data_monitoring_pltb_unidha/DataMonitoring.py

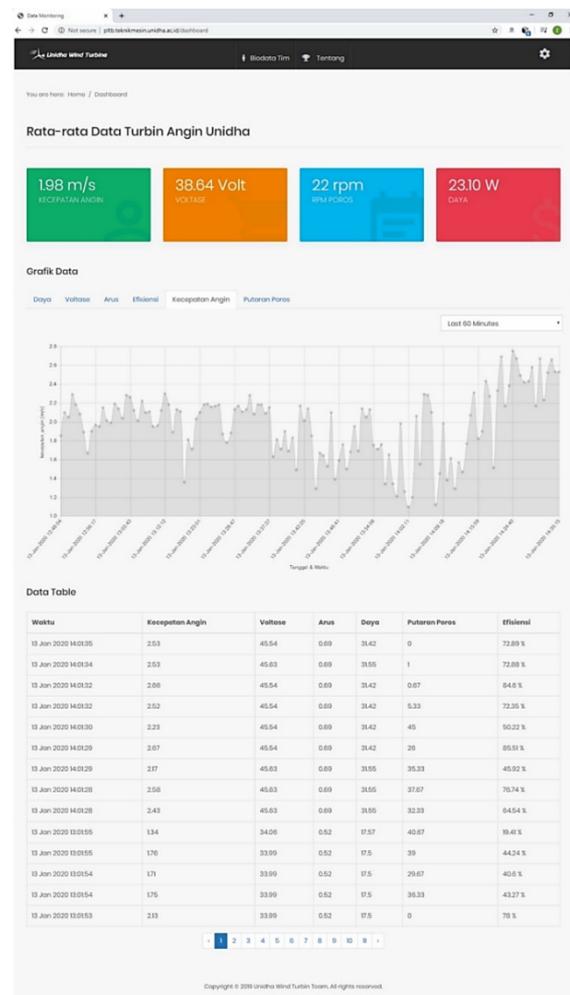
3.2. Website Monitoring Data Turbin Angin

Website sistem monitoring data turbin angin telah berhasil dibuat menggunakan *framework Laravel 5.8* dan *library JavaScript Chart.js* untuk menampilkan data yang dikirim oleh mikrokontroler ke server. Website dibuat dengan menerapkan konsep arsitektur *MVC* pada *framework Laravel 5.8*. Website juga telah di unggah ke server, telah diujicoba, dan dapat diakses pada alamat <http://pltb.teknikmesin.unidha.ac.id> selama proses

pengujian. Ketika alamat website tersebut di akses akan menampilkan informasi awal terkait dengan sistem informasi monitoring data turbin angin dan terdapat tombol akses ke halaman *dashboard* (Gambar 13). Pada halaman *dashboard* telah berhasil menampilkan informasi terkait data turbin angin secara *real-time* dalam bentuk grafik dan tabel (Gambar 14).



Gambar 13. Halaman Depan Website Monitoring Data Turbin Angin



Gambar 14. Halaman Dashboard Website Monitoring Data Turbin Angin

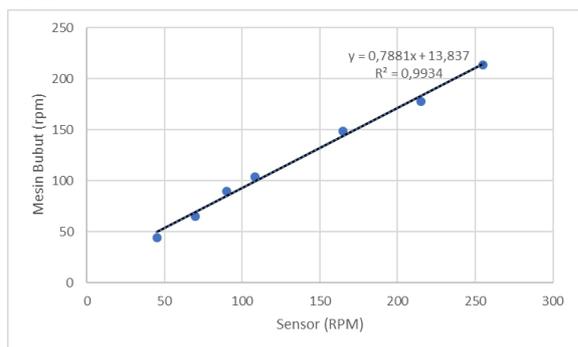
3.3. Hasil Kalibrasi Sensor

Setelah dilakukan kalibrasi pada sensor *IR LM393* diperoleh data hasil pengujian seperti terlihat pada Tabel 1.

Tabel 1. Data Hasil Kalibrasi Sensor *IR LM393*

No.	Putaran Poros (RPM) Rata-rata	
	Mesin Bubut	Sensor
1	45	44
2	70	65
3	90	90
4	108	104
5	165	149
6	215	178
7	255	214

Kalibrasi dilakukan pada variasi rentang kecepatan antara 45 rpm hingga 255 rpm pada mesin bubut. Data pada Tabel 1 di ubah kedalam bentuk grafik dan dicari persamaan regresi liniernya menggunakan *Microsoft Excel* (Gambar 15), sehingga diperoleh persamaan regresi linier pada grafik tersebut adalah $y = 0,7881x + 13,837$ dan nilai $R^2 = 0,9934$. Dimana nilai x adalah nilai kecepatan putaran yang dihasilkan oleh sensor dan y adalah kecepatan putaran pada mesin bubut. Dengan nilai $R^2 = 0,9934$ yang mendekati 1 menunjukkan bahwa nilai variabel x dan y dari proses kalibrasi sensor tersebut berkorelasi dengan baik.



Gambar 15. Grafik Hasil Kalibrasi Sensor *IR LM393*

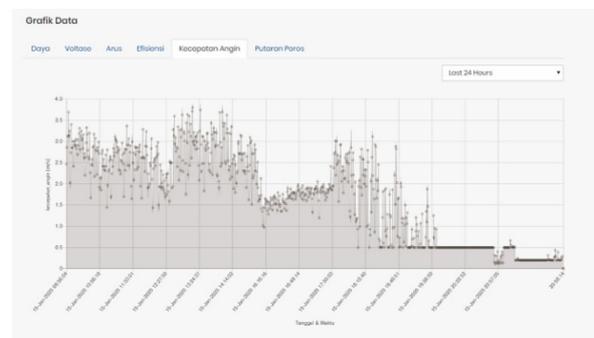
3.4. Hasil Pengujian Website Monitoring Data Turbin Angin

Pengujian dilakukan dengan memasang perangkat turbin dan mikrokontroler di lantai 5 gedung C

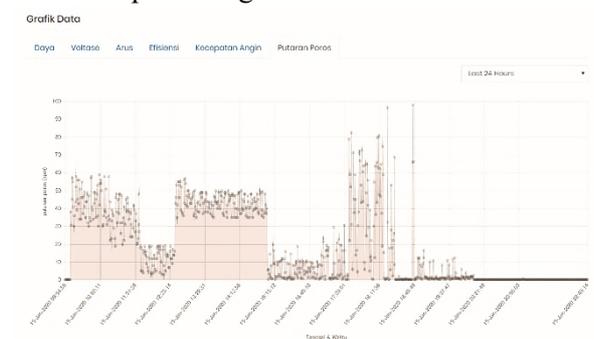
Universitas Dharma Andalas yang berlokasi di Jl. Sawahan No.103A Simpang Haru, Kota Padang. Pengujian dilakukan selama 1 x 24 jam. Raspberry Pi 3B+ dihubungkan dengan koneksi internet melalui perantara modem *GSM*. Berikut hasil pengujian monitoring data turbin angin yaitu:

1. Sensor dan mikrokontroler terkoneksi dengan baik ke server web dan dapat ditampilkan dengan baik. Data dikirim secara *real-time* yang telah dibaca oleh sensor dan dikirim oleh mikrokontroler ke server.
2. Website monitoring data turbin angin dapat menampilkan perubahan data pada grafik secara *live/real-time* untuk setiap variabel data sensor dan berdasarkan katogori waktu yaitu per jam, per hari.

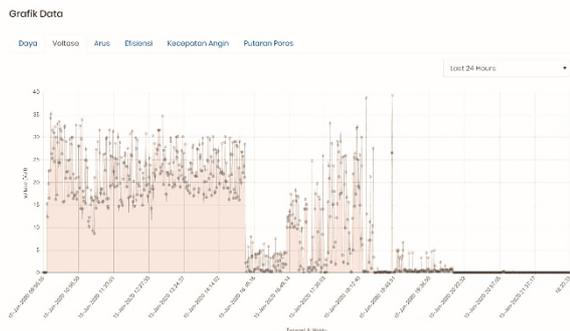
Grafik hasil pengujian awal pada website data monitoring PLTB yang dilakukan selama 24 jam dapat dilihat pada grafik data hasil monitoring kecepatan angin (Gambar 16), grafik data hasil monitoring kecepatan putaran poros (Gambar 17), grafik data hasil monitoring voltase (Gambar 18), grafik data hasil monitoring arus (Gambar 19), grafik data hasil monitoring daya turbin (Gambar 20).



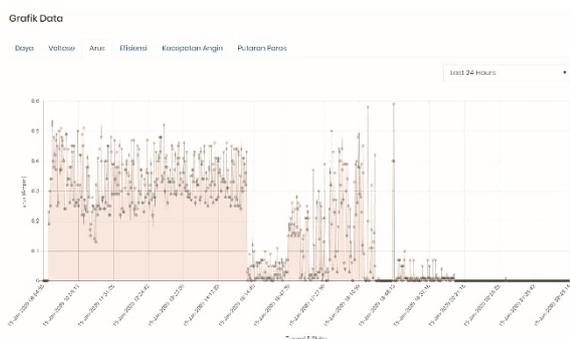
Gambar 16. Grafik Hasil Monitoring Data Kecepatan Angin Turbin melalui Website



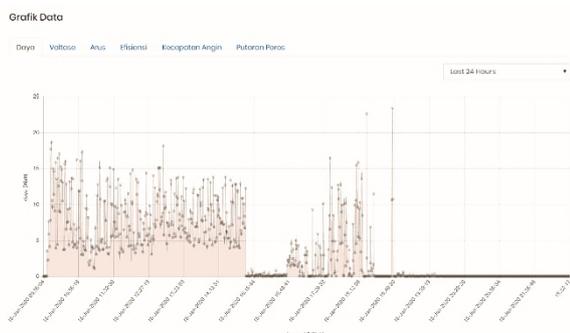
Gambar 17. Grafik Hasil Monitoring Data Kecepatan Putaran Poros Turbin melalui Website



Gambar 18. Grafik Hasil Monitoring Data Voltase Generator melalui Website



Gambar 19. Grafik Hasil Monitoring Data Arus Generator melalui Website



Gambar 20. Grafik Hasil Monitoring Data Daya Generator melalui Website

Berdasarkan hasil pengujian tersebut diperoleh bahwa sistem data monitoring telah berfungsi dengan baik mulai dari sensor dan mikrokontroler, hingga website data monitoring turbin angin. Dimana pada hasil pengujian awal tersebut turbin angin mulai beroperasi pukul 09.54 WIB dengan kecepatan angin yang terbaca oleh sensor adalah 3,53 m/s. Potensi angin tertinggi terjadi pada pukul 13.00 hingga 14.00 WIB dengan kecepatan 3,74 m/s. Mulai pukul 18.00 WIB angin mulai berpindah arah yang semula angin darat kemudian

menjadi angin laut. Kecepatan angin yang terukur pada waktu tersebut berkisar antara 0,5 m/s-1,5 m/s.

4. KESIMPULAN DAN SARAN

Sistem kontrol dan data monitoring turbin angin berbasis website menggunakan Raspberry Pi 3B+ telah berhasil dibuat dan dilakukan pengujian. Berdasarkan hasil pengujian tersebut dapat diambil:

1. Penelitian ini berhasil merancang *website* sistem *monitoring* dan *collecting* data berbasis Raspberry Pi 3B+ yang dapat merekam dan memonitor data turbin angin menggunakan *anemometer*, *IR LM393 speed sensor* dan *INA219 current voltage sensor*.
2. Penelitian ini juga telah berhasil membuat *website monitoring* data turbin angin yang dapat diakses pada alamat <http://pltb.teknikmesin.unidha.ac.id>. Website ini dibuat menggunakan bahasa pemrograman *PHP* menggunakan *framework* *Laravel 5.8* dan *Chart.js* sebagai library yang berfungsi untuk menampilkan grafik *real-time* data turbin angin yang dapat diakses di mana saja.
3. Berdasarkan hasil pengujian kalibrasi sensor *IR LM393* diperoleh persamaan regresi linier $y = 0,7881x + 13,837$ dan nilai $R^2 = 0,9934$. Dan nilai $R^2 = 0,9934$ yang mendekati 1 menunjukkan bahwa proses kalibrasi sensor tersebut terkorrelasi dengan baik.
4. Berdasarkan hasil pengujian monitoring data turbin angin, diperoleh potensi angin tertinggi terjadi pada pukul 13.00 hingga 14.00 WIB dengan kecepatan angin 3,74 m/s.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Ristek Dikti yang telah mendanai penelitian ini dalam program Hibah Penelitian Dosen Pemula (PDP) dengan nomor kontrak: 4/KONTRAK-PENELITIAN-DIKTI/UNIDHA/2020.

DAFTAR PUSTAKA

- [1] Abdillah, M. Hanif, "Sistem Monitoring Secara Real-Time Penyimpanan Energi Listrik dari Wind Turbine Lentera Angin Nusantara (LAN)," in *e-Proceeding of Engineering*, vol. 2, No.2, Agustus, 2015, pp. 6387-6394.
- [2] R.Y. Pratama dan M. Yuhendri, 2020. "Monitoring Turbin Angin Menggunakan Smartphone Android," *Jurnal Teknik Elektro dan Vokasional (JTEV)*, Vol. 6, No. 02, pp. 64-71, 2020.
- [3] D. Handarly dan J. Lianda, "Sistem Monitoring Daya Listrik Berbasis IoT (Internet of Thing)," *Journal of Electrical Electronic Control and Automotive Engineering (JEECAE)*, vol. 3 no. 2, pp. 37-41, November, 2018.
- [4] Winasis, "Desain Sistem Monitoring Nirkabel Berbasis Website Untuk Pemantauan Baterai Dan Beban Pembangkit Listrik Hibrida Surya – Angin," *Jurnal Teknik Elektro (JTE-ITP)*, vol. 5, no. 2, pp 137-142, Juli, 2016.
- [5] F. Shariff, "Photovoltaic remote monitoring system based on GSM," in *IEEE Conference on Clean Energy and Technology (CEAT)*, Langkawi, 18-20 Nov. 2013, pp. 379 – 383.
- [6] L. Wang and K. H. Liu, "Implementation of a Web-Based Real-Time Monitoring and Control System for a Hybrid Wind-PV-Battery Renewable Energy System," in *International Conference on Intelligent Systems Applications to Power Systems (ISAP)*, 2007, pp. 1 – 6.
- [7] C. Kopacz, "Remote and Centralized Monitoring of PV Power Plants," presented at *International Conference on Optimization of Electrical and Electronic Equipment OPTIM*, Brasov, Romania, 2014.
- [8] S. Aryuanto, dkk, "Web-SCADA for Monitoring and Controlling Hybrid Wind-PV Power System". *TELKOMNIKA*, vol. 12, No. 2, June 2014, pp. 305~314.
- [9] P. Y. Muck and M. J. Homam, "Iot Based Weather Station Using Raspberry Pi 3", *International Journal of Engineering & Technology*, vol 7. No.4.30, pp. 145-148, 2018.
- [10] A. Jadhav, "Implementation Of Web Application Using Laravel Framework". *International Journal of Engineering Sciences & Management Research (IJESMR)*, vol. 4, no. 5, pp. 47-51. May, 2017.
- [11] M. Stonebraker, "The 8 Requirements of Real-Time Stream Processing", *SIGMOD Record*, vol. 34, no. 2005.

NOMENKLATUR

NS	Kecepatan putaran poros
Na	Jumlah Celah yang terbaca oleh sensor
Nk	Jumlah celah pada <i>disk encoder</i>
Pr	Periode penghitungan jumlah celah