

Implementasi IOT Node Processing Pada Kasus Monitoring Angin Kencang

1st Faisal Akbar Tasmara

Fakultas Teknik Elektro
Universitas Telkom

Bandung, Indonesia

tasmaraakbar@student.telkomuniversity.ac.id

2nd Ida Wahidah Hamzah

Fakultas Teknik Elektro
Universitas Telkom

Bandung, Indonesia

wahidah@telkomuniversity.ac.id

3rd Fardan

Fakultas Teknik Elektro
Universitas Telkom

Bandung, Indonesia

Fardanfn@telkomuniversity.ac.id

Abstrak - Angin adalah pergerakan udara dari daerah yang bertekanan tinggi ke daerah yang bertekanan rendah. Pembentukan arah angin terjadi karena perbedaan tekanan udara di dua tempat berbeda. Angin kencang adalah salah satu bencana alam yang sering terjadi di Indonesia yang dapat menimbulkan berbagai macam jenis kerusakan atau bahkan timbulnya korban jiwa. Oleh karena itu, mitigasi mengenai bahaya dari angin kencang sangat diperlukan oleh masyarakat. Pengembangan metode klasifikasi diperlukan untuk mendapatkan hasil klasifikasi yang tepat dan cepat untuk mengurangi kesalahan informasi pada masyarakat.

Pada Tugas Akhir ini dirancang sebuah sistem pemantauan angin kencang di beberapa titik dengan implementasi perangkat IOT untuk mendeteksi tingkat bahaya dari angin kencang. Penelitian ini menggunakan berbagai macam jenis parameter seperti kecepatan angin, arah angin, suhu, dan kelembaban udara. Sistem ini diintegrasikan dengan website untuk memudahkan pengguna untuk memantau angin kencang di sekitar node secara real time.

Pada hasil penelitian mengenai alat pendeteksi angin kencang ini menggunakan *simple moving average* untuk melihat fluktuatif pembacaan sensor dan dibandingkan dengan anemometer portable, diperoleh hasil pengujian akurasi alat pendeteksi angin pada node 1 mendapatkan rata-rata akurasi sebesar 95,24% dan pada node 2 mendapatkan hasil akurasi alat sebesar 95,99%.

Kata Kunci: Mitigasi, IoT, Angin Kencang, *Simple Moving Average*

I. PENDAHULUAN

Mitigasi bencana merupakan serangkaian upaya untuk mengurangi dampak bencana sebelum bencana terjadi, termasuk kesiapan dan tindakan-tindakan untuk mengurangi risiko pada jangka waktu yang lama. Mitigasi mengenai angin kencang di Indonesia masih sangat jarang dilakukan oleh pemerintah kepada masyarakat Indonesia termasuk salah satu negara yang minat literasinya sangat rendah. UNESCO telah melakukan riset pada seperti: pertanian, militer, tempat wisata, dan lain sebagainya. Berdasarkan kegiatan-kegiatan tersebut, maka perlu dilakukan pemantauan angin kencang sebagai langkah antisipasi awal untuk memperkecil dampak seperti kecelakaan kerja, atau bahkan timbulnya korban jiwa. tahun 2012 terhadap indeks minat baca di Indonesia hanya 0,001 yang berarti dari 1000 orang di Indonesia hanya satu orang yang suka membaca[1]. Dalam Undang-Undang Nomor 24 Tahun 2007 tentang Penanggulangan Bencana, angin puting beliung didefinisikan sebagai angin kencang yang datang secara tiba-

tiba, mempunyai pusat, bergerak melingkar menyerupai spiral dengan kecepatan 40-50 km/ jam hingga menyentuh permukaan bumi dan akan hilang dalam waktu singkat (3-5 menit)[2].

Pemanfaatan dari pembuatan monitoring angin kencang memiliki manfaat yang sangat penting di berbagai bidang seperti: pertanian, militer, tempat wisata, dan lain sebagainya. Berdasarkan kegiatan-kegiatan tersebut, maka perlu dilakukan pemantauan angin 2 kencang sebagai langkah antisipasi awal untuk memperkecil dampak seperti kecelakaan kerja, atau bahkan timbulnya korban jiwa.

Berikut adalah solusi yang paling mudah digunakan yaitu dengan cara membuat sebuah monitoring angin kencang menggunakan alat yang siap digunakan dan sudah tersedia di toko online. Namun, solusi tersebut membutuhkan biaya yang sangat besar, sehingga kami memutuskan untuk membuat sebuah solusi yang mampu digunakan untuk memonitoring angin kencang dengan mudah. Sistem yang dirancang yaitu dengan membangun node jaringan yang terdiri dari sensor untuk mendeteksi kecepatan angin, arah angin, suhu dan kelembaban udara yang dihubungkan dengan jaringan mesh sampai data bisa dikirim kepada *database* dan dapat dilihat pada website secara *realtime*.

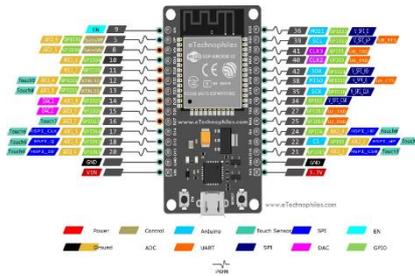
II. KAJIAN TEORI

Pada topik ini, implementasi *IOT Node Processing* digunakan untuk memproses sensor-sensor yang terhubung dengan mikrokontroler ESP32 sehingga data-data dari sensor dapat diolah untuk mendapatkan parameter-parameter yang diukur seperti kecepatan angin, arah angin, suhu, dan kelembaban. Untuk memprogram mikrokontroler ESP32 penulis menggunakan *visual studio code* dan bahasa pemrograman c dengan berbagai macam library, setelah data diproses maka data akan dikirim via MQTT menuju Node-RED, pada Node-RED data yang diterima akan dikirim menuju database *firebase*, sehingga data bisa tersimpan dan dapat ditampilkan pada *website*.

A. ESP 32

ESP32 merupakan sebuah mikrokontroler SoC yang tangguh dengan Wi-Fi 802.11 b/g/n yang terintegrasi, Bluetooth dengan mode ganda versi 4.2, dan berbagai periferal lainnya. Mikrokontroler ini merupakan pengembangan dari chip 8266 terutama dalam hal implementasi dua core clock dengan versi yang berbeda, yaitu hingga 240 MHz. Selain fitur-fitur tersebut, ESP32 juga

terdapat peningkatan jumlah GPIO dari 17 pin menjadi 36 pin, jumlah saluran PWM sebesar 16, dan terdapat memori flash dengan ukuran 4MB.



GAMBAR 1
ESP32

Pada gambar 1 merupakan perangkat ESP32 yang digunakan sebagai mikrokontroler untuk memproses data yang didapatkan dari sensor, dan mengirimkan ke *database*.

B. Sensor Hall effect KY-003

Modul sensor *hall effect* KY-003 merupakan modul yang menggunakan medan magnet untuk dapat beroperasi. Sensor memberikan nilai 1 jika didekatkan dengan medan magnet dan bernilai 0 jika tidak terdapat medan magnet disekitar sensornya, yang akan di proses pada ESP32 agar bisa menghitung RPM (*Revolution Per Minute*) untuk mendeteksi kecepatan angin dan bisa mendeteksi arah datangnya angin.



GAMBAR 2
Hall Effect KY-003

C. Sensor DHT22

Sensor DHT22 merupakan sebuah sensor yang dapat melakukan pendeteksian suhu dan kelembaban baik di dalam maupun di luar ruangan. Sensor DHT22 juga akan diimplementasikan dengan ESP32.



GAMBAR 3
Sensor DHT22

D. Baterai 18650

Untuk daya ESP32 menggunakan baterai lithium-ion 18650, Baterai ini merupakan salah satu baterai yang dapat di cas ulang (*rechargeable*). Nama baterai 18650 merupakan nama yang diperoleh berasal dari ukuran baterai. 18 merupakan diameter baterai yaitu 18mm, 650 yaitu tinggi dari baterai 65,0mm.



GAMBAR 4
Baterai 18650

E. Modul TP4056

Untuk mengisi daya baterai lithium-ion 18650 menggunakan modul *charger* TP4056, dan sumber daya untuk mengisi baterai menggunakan listrik dari panel surya mini 5V. TP4056 merupakan salah satu modul yang dapat digunakan untuk mengisi baterai isi ulang lithium 1 ampere. Pada modul ini dilengkapi dengan lampu indikator yang digunakan untuk mengetahui status dari baterai, pada saat lampu indikator berwarna merah menandakan baterai sedang di isi ulang dan warna biru menandakan baterai sudah terisi penuh.



GAMBAR 5
Modul TP4056

F. Panel Surya Mini 5V

Panel surya mini 5V merupakan salah satu perangkat panel surya yang terkecil yang dapat digunakan untuk mengubah radiasi sinar matahari menjadi energi listrik. Panel surya ini juga dapat digunakan untuk menyalakan berbagai macam peralatan dengan daya yang rendah, seperti perangkat IoT.



GAMBAR 6
Panel Surya Mini 5V

G. Visual Studio Code

Visual studio Code merupakan salah satu software yang digunakan untuk melakukan pengembangan seperti menulis, mengedit, melakukan debug dan membuat kode tertentu. Pada visual studio juga terdapat beberapa fitur seperti pengeditan, pengecekan kesalahan kode, kompilator kode, ekstensi dan masih banyak lagi fitur yang dapat digunakan.

H. MQTT

MQTT atau *Message Queuing Telemetry Transport* adalah protokol transport dengan karakteristik terbuka dan ringan yang dirancang khusus komunikasi untuk *machine-to-machine* dan IOT dengan fitur *publish* dan *subscribe*.

I. Node-RED

Node-RED merupakan software untuk menambungkan alat, API (Application Programming Interface), dan kebutuhan layanan untuk IoT, pemograman node-red berbentuk flow dengan beragam fitur *pallette*, sehingga dapat memudahkan dalam merancang flow untuk sistem Iot.

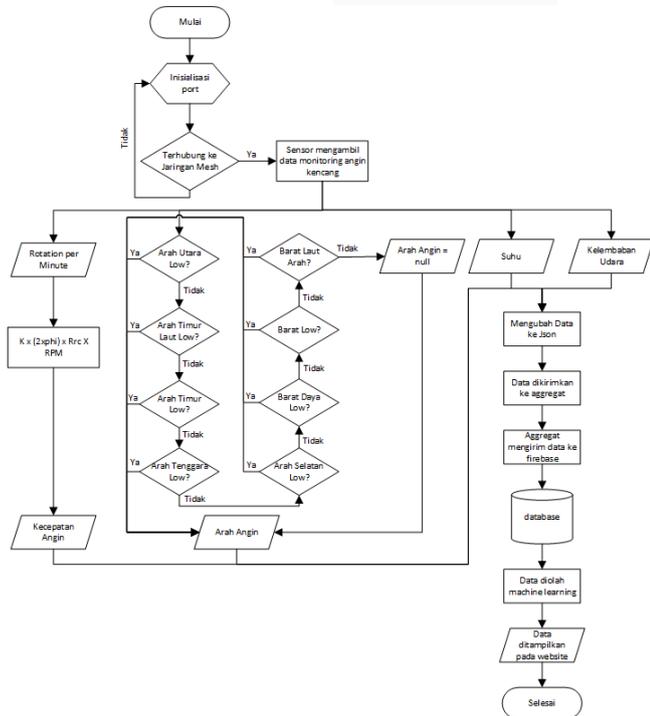
J. SMA (Simple Moving Average)

SMA merupakan filter yang mudah digunakan dan dipahami, Untuk memperhalus pembacaan dari sensor *simple moving average* (SMA) sangat optimal untuk mengurangi *noise*.

III. METODE

Penelitian yang dilakukan yaitu mengenai akurasi alat Pengujian akan dilakukan dengan cara melakukan perbandingan data kecepatan angin antara anemometer uji dengan anemometer portabel dengan cara disandingkan di depan kipas angin yang memiliki 3 buah *mode* kecepatan. Alat akan dilakukan pengujian masing-masing selama 1 menit tiap sesi untuk mendapatkan data kecepatan angin. Untuk pengujian arah angin pengujian dilakukan untuk mengetahui keakuratan sensor pada saat mendeteksi arah angin yang telah dibuat dalam membaca nilai dari arah datangnya angin. Pengujian ini bertujuan untuk melihat rentang perbedaan antara acuan berupa protaktor dengan nilai sudut yang diperoleh dari arah penunjuk angin.

A. Desain Sistem IOT



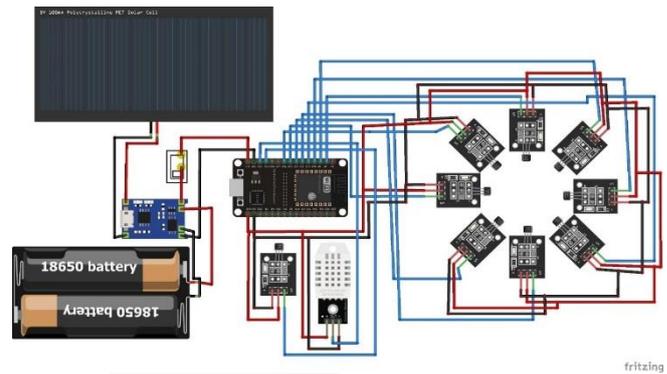
GAMBAR 7
Cara Kerja Node

Berdasarkan gambar 7 *node* akan mengambil data setelah *node* menyala dan terhubung pada jaringan mesh, data yang diambil oleh *node* diantaranya data kecepatan angin, arah angin, suhu dan kelembaban, untuk pengamilan data kecepatan angin bekerja dengan cara menghitung RPM (*Revolution Per Second*) dan mengalikan nilai RPM dengan rumus 1, untuk cara kerja alat penunjuk arah angin

menggunakan logika jika ada *input* dari salah satu sensor maka data arah angin akan disimpan dari inputan sensor yang ada inputan tersebut, jika tidak ada inputan sama sekali maka nilai atau data arah angin sama dengan *NULL*, dari keempat data tersebut selanjutnya digabungkan ke dalam *format* JSON (*JavaScript Object Notation*) agar mudah dikirim sekaligus dengan contoh seperti berikut {"Node": "1", "Kecepatan": "0.00", "Arah": "Selatan", "Temperature": "20.11", "Humidity": "80.12"}. Setelah data dalam bentuk JSON maka data akan dikirim pada *node* *aggregat* untuk dikirim ke database, agar website bisa mengambil data dari database untuk ditampilkan.

Pada gambar 4 merupakan hasil implementasi *painlessmesh*, pada terdiri dari satu buah laptop yang berfungsi untuk sumber daya dari *aggregat* dan *root node*. Pada pengujian terdiri dari 3 buah *node*, dimana pada *node* 1 dan *node* 2 menggunakan mikrokontroller ESP32, sensor *hall effect* untuk mendeteksi kecepatan angin dan arah angin serta menggunakan DHT22 untuk mendeteksi suhu dan kelembaban udara. Pada *node* 3 terdiri dari mikrokontroller ESP32 dan DHT22, seluruh *node* dihubungkan dengan menggunakan jaringan *WiFi mesh*.

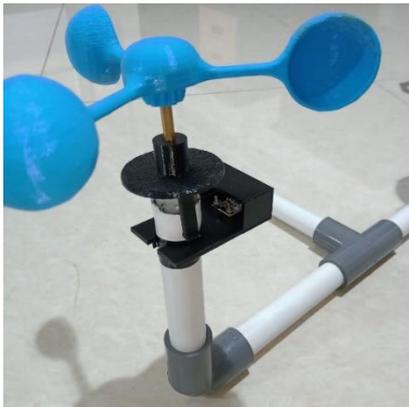
B. Implementasi Perangkat IOT



GAMBAR 8
Skematik Perangkat IOT

Berdasarkan gambar 8 skematik tersebut menggunakan 1 mikrokontroller ESP32 8 sensor *hall effect* untuk alat penunjuk arah angin, 1 sensor *hall effect* untuk menghitung RPM untuk anemometer, sensor DHT22 untuk mendeteksi suhu udara dan kelembaban, baterai 18650, modul TP4056 dan panel surya mini 5V sebagai *power supply* untuk mikrokontroller dan sensor-sensor yang terhubung.

1. Implementasi sensor *hall effect* untuk deteksi RPM pada anemometer



GAMBAR 9
Implementasi pada anemometer cup

Berdasarkan gambar 10 adalah hasil dari implementasi sistem pendeteksi RPM untuk menghitung kecepatan angin menggunakan sensor *hall effect* yang digabungkan dengan anemometer *cup* dan magnet neodim sebagai inputan untuk sensor, dengan cara kerja setiap cup berputar maka magnet neodim akan ikut berputar, setiap perputaran magnet akan memberikan input pada sensor *hall effect*, nilai inputan berupa jumlah putaran magnet yang terhitung oleh sensor, nilai putaran tersebut akan proses oleh ESP32 untuk mendapatkan nilai RPM dan kecepatan angin, untuk mendapatkan nilai RPM menggunakan rumus 1 di bawah:

$$RPM = \frac{\text{Jumlah Putaran} \times 60}{\text{Waktu Pengamatan}} \quad (1)$$

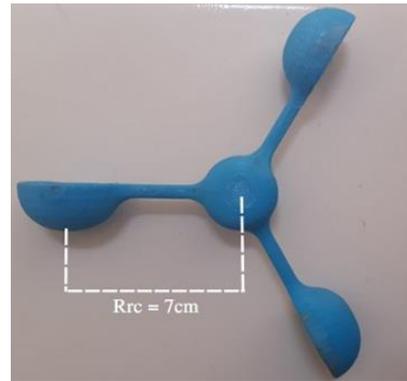
Untuk menghitung kecepatan angin dengan anemometer cup, Ketika sebuah *cup* membuat satu putaran penuh, *cup* menempuh jarak yang sama dengan keliling lingkaran yang berjari-jari 9 cm dengan rumus $2 \times 3,1416 \times 9 \text{ cm}$ [3]. Tetapi karena jenis cup anemometer yang di pakai berpengaruh pada hasil perhitungan kecepatan angin maka digunakan persamaan 2 di bawah[4].

$$V = K \times (2 \times \pi) \times f \times Rrc \quad (2)$$

Keterangan:

- V : Kecepatan angin (m/s)
- K : Faktor anemometer
- f : Frekuensi (hz)
- Rrc : Jarak dari pusat anemometer ke titik pusat *cup* (cm)

Untuk R_{rc} merupakan panjang lengan ke titik pusat pada cup anemometer[Ref 8 dibuka].



GAMBAR 10
Rrc Pada cup untuk anemometer

Anemometer faktor (*K*) yang berpengaruh berdasarkan karakteristik *cup* di mana *K_d* merupakan hubungan koefisien *drag* yang tidak berdimensi antara *cup* depan dan belakang, di mana standar *C_d* (*coefficient drag*) tergantung dari bentuk *cup* anemometer, karena yang dipakai pada pengujian ini adalah *cup* anemometer setengah lingkaran, maka nilai *C_d* adalah *C_{d1}*=1.42, dan *C_{d2}*=0.38[5], untuk menghitung faktor anemometer dapat dilihat pada persamaan 3 dan 4 dibawah:

$$K = \frac{kd+1}{kd-1} \quad (3)$$

$$kd = \sqrt{\frac{cd1}{cd2}} \quad (4)$$

Keterangan:

- K : Faktor anemometer
- cd : *coefficient drag*
- K_d : Hubungan koefisien drag dari *cup* anemometer

Untuk mendapatkan nilai f (frekuensi) didapat dari persamaan 5 dibawah :

$$f = \frac{RPM}{60} \quad (5)$$

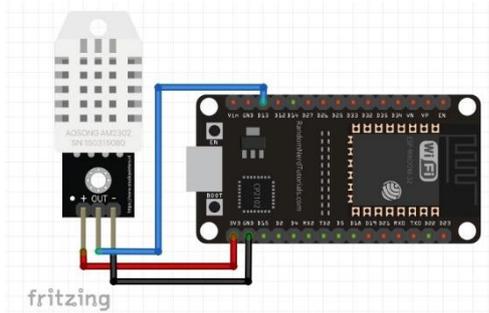
2. Impelemtasi sensor *hall effect* untuk deteksi arah angin



GAMBAR 11

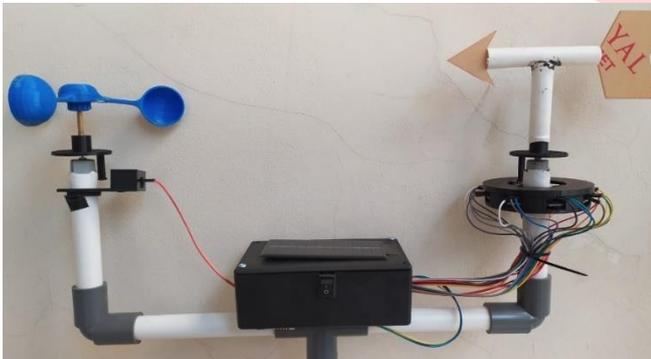
Implementasi Pendeteksi Arah Angin Menggunakan Sensor *Hall effect*
Pada gambar 13 menggunakan desain 3d untuk tempat penyimpanan 8 buah sensor *hall effect* untuk mendeteksi arah angin, di mana setiap sensor *hall effect* melambangkan satu arah mata angin. Arah mata angin terdiri dari utara, timur

laut, timur, tenggara, selatan, barat daya, barat, dan barat laut. Pada gambar 13 merupakan skema dari pendeteksi arah angin dengan 8 sensor *hall effect* dengan dengan pin *input* dari sensor *hall effect* terhubung ke GPIO36,39,34,35,32,33,25,26 pada ESP32.



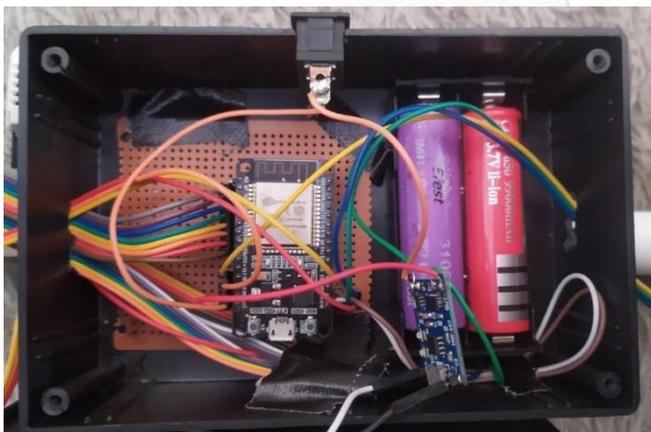
GAMBAR 12 Implementasi DHT22 dan ESP32

Pada gambar 14 merupakan skema DHT22 dan ESP32 dengan *input* pin DHT22 terhubung dengan GPIO13 pada ESP32.



GAMBAR 13 Hasil Implementasi Sebuah Node

Gambar 15 merupakan hasil implementasi dari sebuah *node* jika tampak dari luar.



GAMBAR 14 Didalam Box 1 Node

Berdasarkan pada gambar 16 merupakan hasil dari implementasi sistem satu *node* jika dilihat dari dalam box. *Node* terdiri dari PCB, ESP32, Modul TP4056, tombol switch, dudukan baterai dan 2 buah baterai 18650.

3. *Simple Moving Average* (SMA)

SMA pada sistem ini di implementasikan pada nilai kecepatan angin dan dengan cara kerja mencari nilai rata-rata dari nilai kecepatan angin dengan menghitungnya menggunakan bahasa pemrograman c. Saat menghitung nilai yang berurutan SMA akan menjumlahkan data yang masuk dan dibagi dengan banyak data, dan jika ada nilai baru yang masuk maka data terlama akan dibuang, SMA dapat dihitung dengan persamaan 6 di bawah[6].

$$SMA = \frac{P_M + P_{M-1} + \dots + P_{M-(n-1)}}{n} \quad (6)$$

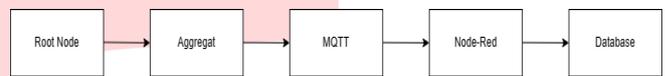
Keterangan :

SMA : *Simple Moving Average*

P_M : Nilai data dari waktu M

n : Banyak data yang digunakan

C. Metode pengiriman data



GAMBAR 15

Diagram Blok Pengiriman dan Penyimpanan Data

Berdasarkan pada gambar 4.16 menunjukkan proses pengiriman data yang dimulai dari *bridge* ataupun *root node* untuk mengirimkan data ke *agregat*. *Agregat* mengirimkan data menggunakan protokol MQTT dan diteruskan ke *node-RED* yang berfungsi untuk menghubungkan antara data MQTT dan database.

D. Pengujian akurasi anemometer cup dengan anemometer portable

Pengujian dilakukan untuk mengetahui keakuratan sensor pada anemometer uji yang telah dibuat dalam membaca nilai dari putaran anemometer *cup*. Pengujian ini ditujukan untuk melihat rentang perbedaan antara acuan berupa anemometer portabel dengan nilai yang diperoleh dari anemometer uji.

TABEL 1 Rumus Akurasi dan Error

No.	Nama Rumus	Rumus
1.	Error (%)	$\frac{(\text{Hasil alat pembanding} - \text{Hasil alat})}{\text{Hasil alat pembanding}} \times 100\%$ (7)
2.	Akurasi (%)	$100\% - \text{Rata-rata error} (\%)$ (8)

IV. HASIL DAN PEMBAHASAN

Pengujian akan dilakukan dengan cara melakukan perbandingan data kecepatan angin antara anemometer uji dengan anemometer portabel dengan cara disandingkan di depan kipas angin yang memiliki 3 buah *mode* kecepatan. Alat akan dilakukan pengujian masing-masing selama 1 menit tiap sesi untuk mendapatkan data kecepatan angin, hasil data yang didapat dihitung presentase *error* dan akuransinya menggunakan rumus dari tabel 1.

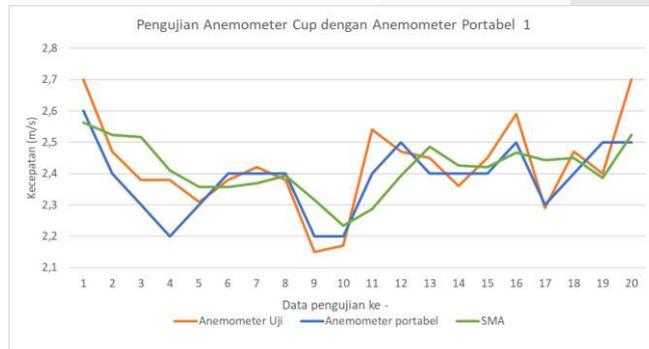
A. Akurasi dan *error* anemometer cup

1. Data hasil Pengujian anemometer cup ke-1

TABEL 2
Hasil Pengujian Alat Uji Dengan Anemometer Portabel 1

Nomor Pengujian	Hasil Pengujian Alat (m/s)	Anemometer Portabel (m/s)	Erorr (%)	SMA (m/s)
1	2,7	2,6	3,84	2,56
2	2,47	2,4	2,91	2,52
3	2,38	2,3	3,47	2,52
4	2,38	2,2	8,18	2,41
5	2,31	2,3	0,434	2,36
6	2,38	2,4	0,833	2,36
7	2,42	2,4	0,833	2,37
8	2,38	2,4	0,833	2,39
9	2,15	2,2	2,272	2,32
10	2,17	2,2	1,363	2,23
11	2,54	2,4	5,833	2,29
12	2,47	2,5	1,2	2,39
13	2,45	2,4	2,083	2,49
14	2,36	2,4	1,67	2,43
15	2,45	2,4	2,083	2,42
16	2,59	2,5	3,6	2,47
17	2,29	2,3	0,434	2,44
18	2,47	2,4	2,91	2,45
19	2,4	2,5	4	2,39
20	2,7	2,5	8	2,52
Rata-rata	2,42	2,38	2,84	2,42

Berdasarkan tabel 2 didapatkan nilai pengujian alat uji dan dibandingkan dengan anemometer portabel dengan rata-rata alat hasil uji 2,42 m/s dan rata-rata anemometer portabel 2,38 m/s dengan rata rata *error* 2,84%. Nilai rata-rata SMA 2,42 m/s dan akurasi mencapai 97,16%.



GAMBAR 16
Grafik Perbandingan Alat Uji Dengan Anemometer Portabel 1

Gambar 16 merupakan grafik perbandingan dari nilai anemometer uji, anemometer portabel sebagai pembanding dan nilai SMA (*simple moving average*) dari pengujian 1 pada *node* 1, dari gambar diatas bisa dilihat bahwa sebagian besar nilai dari anemometer uji jika dibandingkan dengan

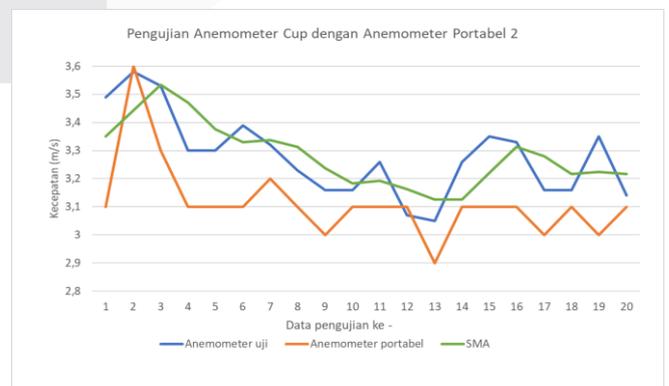
anemometer portabel selisihnya sangat kecil ,dengan selisih rata-rata 0,038, untuk nilai dari SMA akan mengikuti 3 nilai anemometer uji sebelumnya jadi nilainya tidak akan jauh berbeda dengan nilai anemometer uji.

2. Data hasil Pengujian anemometer cup ke-2

TABEL 3
Hasil Pengujian Alat Uji Dengan Anemometer Portabel 2

Nomor Pengujian	Hasil Pengujian Alat (m/s)	Anemometer Portabel (m/s)	Erorr (%)	SMA (m/s)
1	3,49	3,1	3,35	2,56
2	3,58	3,6	3,44	2,52
3	3,53	3,3	3,53	2,52
4	3,3	3,1	3,47	2,41
5	3,3	3,1	3,38	2,36
6	3,39	3,1	3,33	2,36
7	3,32	3,2	3,34	2,37
8	3,23	3,1	3,31	2,39
9	3,16	3	3,24	2,32
10	3,16	3,1	3,18	2,23
11	3,26	3,1	3,19	2,29
12	3,07	3,1	3,16	2,39
13	3,05	2,9	3,13	2,49
14	3,26	3,1	3,13	2,43
15	3,35	3,1	3,22	2,42
16	3,33	3,1	3,31	2,47
17	3,16	3	3,28	2,44
18	3,16	3,1	3,22	2,45
19	3,35	3	3,22	2,39
20	3,14	3,1	3,22	2,52
Rata-rata	3,28	3,12	3,28	2,42

Berdasarkan tabel 3 didapatkan nilai pengujian alat uji dan dibandingkan dengan anemometer portabel dengan rata-rata hasil alat uji 3,28 m/s dan rata-rata anemometer portabel 3,12 m/s dengan rata rata *error* 5,43 %. Nilai rata-rata SMA 3,28 m/s dan akurasi sebesar 94,57%.



GAMBAR 17
Grafik Perbandingan Alat Uji Dengan Anemometer Portabel 2

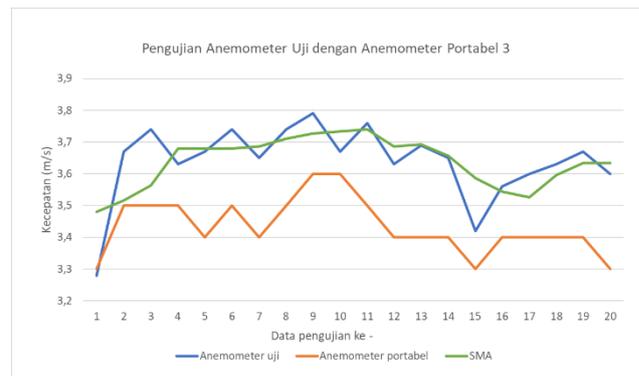
Gambar 17 merupakan grafik perbandingan dari nilai anemometer uji, anemometer portabel sebagai pembandingan dan nilai SMA (simple moving average) dari pengujian ke 2 pada node 1, dari gambar diatas bisa dilihat bahwa nilai dari anemometer uji jika dibandingkan dengan anemometer portabel selisihnya terlihat cukup signifikan perbedaannya ,dengan selisih rata-rata 0,16, untuk nilai dari SMA akan mengikuti 3 nilai anemometer uji sebelumnya jadi nilainya tidak akan jauh berbeda dengan nilai anemometer uji.

3. Data hasil Pengujian anemometer cup ke-3

TABEL 4
Hasil Pengujian Alat Uji Dengan Anemometer Portabel 3

Nomor Pengujian	Hasil Pengujian Alat (m/s)	Anemometer Portabel (m/s)	Error (%)	SMA (m/s)
1	3,28	3,3	0,60	3,48
2	3,67	3,5	4,85	3,52
3	3,74	3,5	6,85	3,56
4	3,63	3,5	3,71	3,68
5	3,67	3,4	7,94	3,68
6	3,74	3,5	6,85	3,68
7	3,65	3,4	7,35	3,69
8	3,74	3,5	6,85	3,71
9	3,79	3,6	5,27	3,73
10	3,67	3,6	1,94	3,73
11	3,76	3,5	7,42	3,74
12	3,63	3,4	6,76	3,69
13	3,69	3,4	8,52	3,69
14	3,65	3,4	7,35	3,66
15	3,42	3,3	3,63	3,59
16	3,56	3,4	4,70	3,54
17	3,6	3,4	5,88	3,53
18	3,63	3,4	6,76	3,60
19	3,67	3,4	7,94	3,63
20	3,6	3,3	9,09	3,63
Rata-rata	3,64	3,44	6,01	3,64

Berdasarkan tabel 4 didapatkan nilai pengujian alat uji dan dibandingkan dengan anemometer portabel dengan rata-rata hasil alat uji 3,64 m/s dan rata-rata anemometer portabel 3,44 m/s dengan rata rata error 6,01 %. Nilai rata-rata SMA 3,64 m/s dengan akurasi alat mencapai 93,99%.



GAMBAR 18

Grafik Perbandingan Alat Uji Dengan Anemometer Portabel 2
Gambar 18 merupakan grafik perbandingan dari nilai anemometer uji, anemometer portabel sebagai pembandingan dan nilai SMA (simple moving average) dari pengujian ke 3 pada node 1, dari gambar diatas bisa dilihat bahwa nilai dari anemometer uji jika dibandingkan dengan anemometer portabel, dari sebagian besar data pengujian diatas selisihnya terlihat agak jauh ,dengan selisih rata-rata 0,21, untuk nilai dari SMA akan mengikuti 3 nilai anemometer uji sebelumnya jadi nilainya tidak akan jauh berbeda dengan nilai anemometer uji.

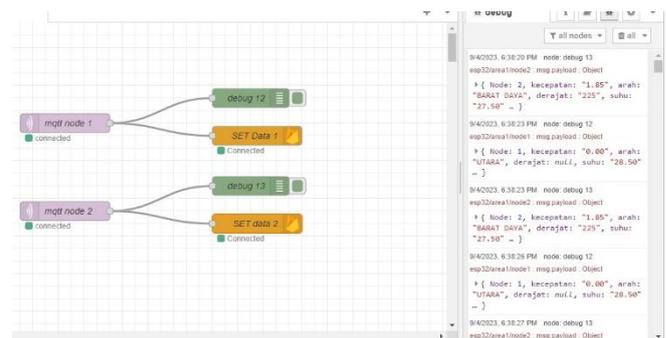
B. Pengujian akurasi alat penunjuk arah angin

TABEL 5
Data Hasil Pengujian Deteksi Arah Angin

No	Arah Angin	Derajat Arah Angin	Derajat Busur	Selisih (%)
1	UTARA	360/0	0	0
2	TIMUR LAUT	45	45	0
3	TIMUR	90	87	3,44
4	TENGGARA	135	140	3,57
5	SELATAN	180	180	0
6	BARAT DAYA	225	220	2,27
7	BARAT	270	270	0
8	BARAT LAUT	315	310	1,61
			Rata-rata	1,55

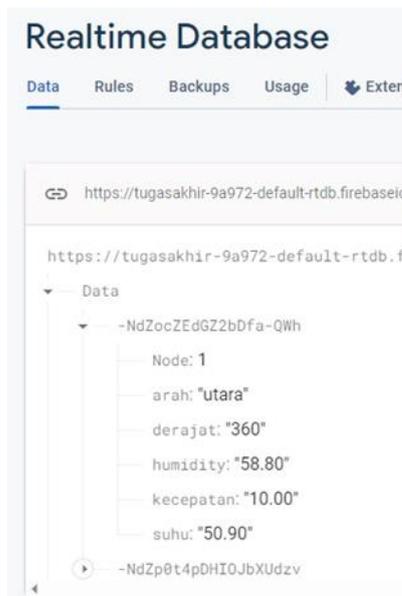
Berdasarkan tabel 5 merupakan hasil pengujian yang telah dilakukan pada pengujian akurasi deteksi arah angin pada jarak 20cm mendapatkan rata-rata akurasi sebesar 98,84% dari setiap kali percobaan terhadap arah datangnya angin.

C. Pengiriman data



GAMBAR 19
Konfigurasi *node-RED*

Pada gambar 19 merupakan aplikasi *node-RED* dengan *node* MQTT terhubung *debug* dan *node* *firebase*, *debug* pada *node-RED* berfungsi sebagai pengecek apakah ada kesalahan pada *node* MQTT, untuk *node* *firebase* berfungsi untuk menyimpan data yang diterima dari *node* MQTT.



GAMBAR 20
Database *firebase*

Pada gambar 20 data dengan format JSON yang dikirim melalui protokol MQTT dan diteruskan melalui *node-RED* dapat diterima tanpa masalah pada *database firebase*.

V. KESIMPULAN

Berdasarkan hasil pengujian anemometer uji dengan pengujian selama 1 menit tiap sesinya dilakukan sebanyak 20 kali dengan menggunakan 3 mode berbeda kecepatan kipas angin, didapatkan hasil pengujian menggunakan kipas angin mode 1 sampai 3 mendapatkan nilai *error* rata-rata yang meningkat seiring bertambahnya kecepatan angin, dan dapat disimpulkan seiring bertambahnya kecepatan angin maka semakin menurun akurasi, dengan rata-rata akurasi 95,24% untuk akurasi alat diatas 90% dapat disimpulkan bahwa alat ukur akurasi baik, pada pengujian akurasi alat penunjuk arah angin pada mendapatkan hasil akurasi sebesar 98,94%, untuk nilai *error* rata-rata dari *simple moving average* sangat mendekati nilai *error* rata-rata dari anemometer uji dan dapat disimpulkan dapat bekerja dengan baik karena *simple moving average* menghitung nilai rata-ratanya saja, untuk pengiriman data melalui protokol MQTT dan *node-RED* dapat berjalan dengan baik dan data dapat terkirim pada *database firebase* tanpa masalah.

REFERENSI

- [1] A. Nafisah, "Arti penting perpustakaan bagi upaya peningkatan minat baca masyarakat," *LIBRARIA: Jurnal Perpustakaan*, vol. 2, no. 2, 2016.
- [2] Mohd. R. Amri, *RBI: risiko bencana Indonesia*. Badan Nasional Penanggulangan Bencana, 2016.
- [3] Science First, *652-1010 (05-005) Anemometer Instructions*. 2005.
- [4] E. Avallone *et al.*, "An inexpensive anemometer using Arduino board," *Facta universitatis - series: Electronics and Energetics*, vol. 32, no. 3, pp. 359–368, 2019, doi: 10.2298/FUEE1903359A.
- [5] Bruce R. Munson, Donald F. Young, Theodore H. Okiishi, and Wade W. Huebsch, *FUNDAMENTALS OF FLUID MECHANICS*, 6th ed. United States Of America: Don Fowley, 2009.
- [6] S. Hansun, "A new approach of moving average method in time series analysis," in *2013 Conference on New Media Studies (CoNMedia)*, 2013, pp. 1–4. doi: 10.1109/CoNMedia.2013.6708545.