bukti_3283-Article_Text-5113-1-10-20210129.pdf

by

Submission date: 14-Feb-2023 08:57AM (UTC+0700)
Submission ID: 2013614154
File name: bukti_3283-Article_Text-5113-1-10-20210129.pdf (739.38K)
Word count: 4832
Character count: 28170

JURNAL TEKNOLOGI TECHNOSCIENTIA
Vol. 14 No. 1 Agustus 2021

ISSN: 1979-8415
E-ISSN: 2714-8025

SISTEM MONITORING KUALITAS AIR TAMBAK UDANG MENGGUNAKAN *REAL TIME OPERATING SYSTEM*

Sabtian Juliana[1], Adnan[2], Christoforus Yohannes[3]
[1,2,3]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin
Email:[1]sabtian8@gmail.com,[2]adnan@unhas.ac.id,[3]christ.mitra@gmail.com

Masuk: 16 Oktober 2020, Revisi masuk: 10 Januari 2021, Diterima: 28 Januari 2021

ABSTRACT

In shrimp farming, pond water quality is a very important factor to be considered by farmers. Some water quality parameters that need to be considered include temperature, salinity, pH, and pond water level. Manual or conventional pond water quality reading systems make handling slow even though shrimp are sensitive animals to changes in water quality. Therefore, in this study a system that can monitor pond water quality is developed using ESP32. ESP32 is a microcontroller successor to the ESP8266 which already has a dual core processor. However, even though you already have a dual core processor, only one core will be used by default. Therefore, the Real Time Operating System (RTOS) is used in the system to maximize the processor it has. Testing is done by sending data every minute and calculating data loss that occurs in the system being created. Tests were carried out 5 times with a distance of 10 meters, 20 meters, 25 meters, 30 meters and 35 meters between the nodes. The results of the data loss testing that occurred at a distance of 10 meters was 0.0%, at a distance of 20 meters 0.0%, at a distance of 25 meters 0.0%, at a distance of 30 meters 0.4%, and at a distance of 35 meters 38.3%.

Keywords: *Data Loss, Esp Mesh, Real Time Operating System, RTOS.*

INTISARI

Dalam pembudidayaan udang, kualitas air tambak merupakan faktor yang sangat penting untuk diperhatikan oleh petambak. Beberapa parameter kualitas air yang perlu diperhatikan diantaranya adalah suhu, salinitas, pH, dan ketinggian air tambak. Sistem pembacaan kualitas air tambak secara manual atau konvensional membuat penanganan menjadi lambat padahal udang merupakan hewan yang sensitif terhadap perubahan kualitas air. Oleh karena itu, dalam penelitian ini dibuatlah sebuah sistem yang dapat memonitoring kualitas air tambak dengan menggunakan ESP32. ESP32 merupakan mikrokontroller penerus ESP8266 yang sudah memiliki dual core prosesor. Akan tetapi, walaupun sudah memiliki prosesor dual core, hanya satu core yang akan digunakan secara default. Oleh karena itu, dalam sistem yang dibuat digunakanlah Real Time Operating System (RTOS) untuk memaksimalkan prosesor yang dimiliki. Pengujian dilakukan dengan cara mengirim data setiap menit dan menghitung data loss yang terjadi pada sistem yang dibuat. Pengujian dilakukan sebanyak 5 kali dengan jarak 10 meter, 20 meter, 25 meter, 30 meter, dan 35 meter antar node. Hasil dari pengujian yang dilakukan data loss yang terjadi pada jarak 10 meter adalah 0,0%, pada jarak 20 meter 0,0%, pada jarak 25 meter 0,0%, pada jarak 30 meter 0,4%, dan pada jarak 35 meter 38,3%.

Kata-kata kunci: *Data Loss, Esp Mesh, Real Time Operating System, RTOS.*

PENDAHULUAN

Internet of Things adalah sebuah konsep dimana sebuah objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa interaksi manusia ke manusia ataupun manusia ke komputer. Sederhananya, *Internet of Things* adalah alat yang terhubung dengan internet dan saling terintegrasi. Fungsi utama *Internet of Things* pada dasarnya sebagai *data miner*. *Internet of Things* bekerja mencari dan mengumpulkan berbagai data yang nantinya akan diolah menjadi informasi yang lebih bermanfaat. *Internet of Things* sangat erat hubungannya dengan revolusi industri 4.0 karena *IoT* adalah unsur utamana dalam revolusi tersebut. *IoT* berpengaruh dalam berbagai macamindustri seperti manufaktur, logistik, tata kota, rumah pertanian, dan lain sebagainya.

Tambak merupakan salah satu jenis habitat yang digunakan untuk kegiatan budidaya air payau yang berada di pesisir, dimana kegiatan budidaya yang dilakukan

ISSN: 1979-8415
E-ISSN: 2714-8025

secara terus menerus dapat menyebabkan terjadinya degradasi terhadap lingkungan dan ditandai dengan menurunnya kualitas air (Muhammuddin, 2015).

Real Time Operating System (RTOS) adalah sistem operasi yang digunakan untuk memenuhi kebutuhan aplikasi yang bersifat *realtime*. *Realtime* disini berarti ia membutuhkan kinerja setiap saat dimana ia dibutuhkan saat itu juga. Salah satu kunci keberhasilan RTOS adalah kemampuannya untuk melakukan kerja secara konsisten baik secara waktu yang ia butuhkan maupun secara task aplikasi yang mampu ia kerjakan.

ESP32 adalah mikrokontroller yang dikenalkan oleh Espressif System dan merupakan penerus dari ESP8266. ESP32 ini sudah memiliki modul WIFI dan Bluetooth didalamnya serta mempunyai CPU Xtensa dual core LX6–160MHz (Espressif Systems (Shanghai) Co., Ltd., 2016)). Meskipun memiliki prosesor dual core tapi secara default hanya satu core digunakan. Maka untuk memaksimalkan kinerja dari ESP32 penulis akan menggunakan RTOS agar kedua core tersebut dapat bekerja mengeksekusi perintah sehingga program akan berjalan lebih efektif.

Rumusan masalah penelitian ini adalah:

1. Bagaimana merancang alat monitoring suhu, kadar garam, pH, dan ketinggian air tambak udang?
2. Apakah ESP32 dapat melakukakn pengambilan data, transformasi data, *routing*, dan komunikasi dalam selang waktu tertentu menggunakan *Real Time Operating System*?
3. Berapa persentase *data loss* yang terjadi pada saat pengiriman data?

Tujuan akhir dari penelitian ini adalah:

1. Untuk membuat sistem yang dapat memonitoring suhu, kadar garam, pH, dan ketinggian air tambak udang.
2. Untuk mengetahui kemampuan ESP32 dalam melakukan pengambilan data, transformasi data, *routing*, dan komunikasi menggunakan *Real Time Operating System.*
3. Untuk mengetahui *data loss* yang terjadi pada saat pengiriman data.

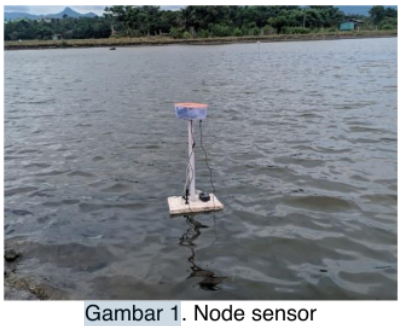
METODE

Waktu penelitian dilakukan sejak September 2019 hingga proses pelaporan hasil penelitian ini pada bulan November 2020. Pengambilan data dilakukan di Kelurahan Mangkoso Kecamatan Soppeng Riaja Kabupaten Barru. Perancangan sistem, evaluasi, dan pengolahan data dilakukan di Laboratorium Internet of Things dan Parallel Computing (IoTPC), Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin. Instrumen penelitian yang digunakan pada penelitian ini meliputi:

1. Software
   - Windows 10 x64
   - Arduino IDE
   - Thingsboard
2. Hardware
   - Laptop, RAM 8GB, Prosesor Intel® Core™ i5-5200 CPU @2.20 GHz
   - Mikrokontroller ESP32
   - Battery Shield
   - Baterai 18650 berkapasitas 12000 mAh
   - Sensor ultrasonic
   - Sensor pH
   - Sensor kadar garam
   - Sensor suhu DS18B20

Data yang digunakan pada penelitian ini yaitu jumlah data yang diambil dari setiap node dan data yang diterima oleh sever Thingsboard. Dari data tersebut akan diolah untuk mendapatkan hasil persentase data loss yang dilakukan oleh ESP32.

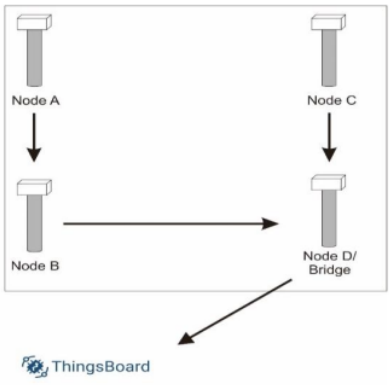
Data yang dikirim oleh setiap sensor merupakan data kualitas air tambak yang diambil dari beberapa sensor yang sudah di transformasikan ke dalam bentuk json. Setiap node akan mengirimkan data setiap 1 detik. Ilustrasi node sensor dapat dilihat pada Gambar 1.



Gambar 1. Node sensor

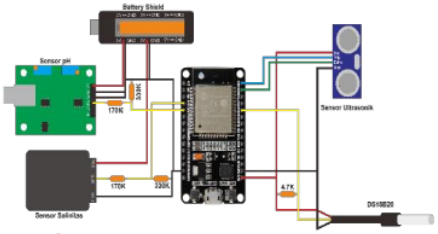
Pengiriman data dalam sistem ini menggunakan topologi mesh, agar alur pengiriman data mudah diawasi maka digunakan static routing. Ilustrasi pengiriman data dapat dilihat pada Gambar 2.

ISSN: 1979-8415
E-ISSN: 2714-8025



Gambar 2. Ilustrasi pengiriman data

Desain perangkat keras sistem yang dibuat seperti Gambar 3.



Gambar 3. Desain perangkat keras

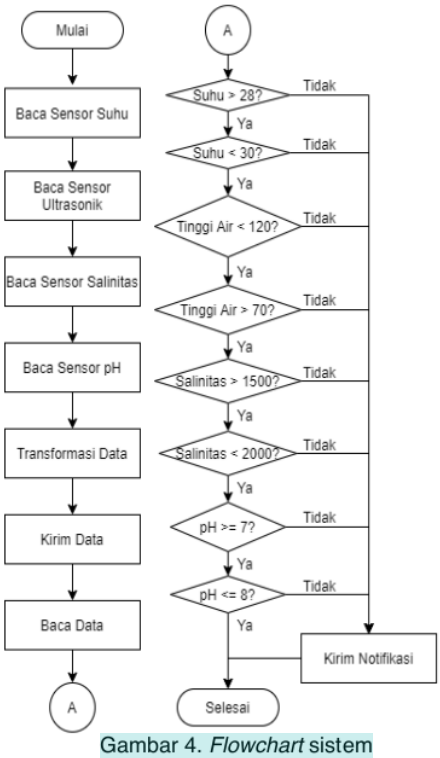
Berdasarkan Gambar 3, sensor pH dan sensor kadar garam mengambil sumber tegangan 5V dari battery shield, hal ini dikarenakan pada ESP32 tidak memiliki pin output 5V. Output sensor pH yaitu pin Po dihubungkan ke pin 34 di ESP32, akan tetapi karena output dari sensor pH adalah 5V sedangkan tegangan kerja pada ESP32 adalah 3.3V maka digunakan resistor 170K diantara Po dan pin 34 dan resistor 330K dari pin 35 ke ground. Pin G dihubungkan ke ground.

Untuk sensor kadar garam outputnya dihubungkan ke pin 35. Karena output dari sensor kadar garam juga 5V maka sama seperti sensor pH digunakan resistor 170K antara pin output dari sensor kadar garam ke pin 34 dan resistor 330K dari pin 35 ke ground. Pin GND dihubungkan ke ground.

Pada sensor ultrasonik Vcc dihubungkan ke pin 3V3 sebagai sumber tegangan, pin trigger dihubungkan ke pin 23, pin echo dihubungkan ke pin 22, dan Gnd dihubungkan ke ground.

Pada sensor suhu DS18B20 Vcc dihubungkan ke pin 3V3. Output dihubungkan pada pin 21. Digunakan resistor 4.7K antara output sensor dan Vcc. Gnd dihubungkan ke ground.

Adapun *flowchart* sistem secara keseluruhan dapat dilihat pada Gambar 4.



Gambar 4. *Flowchart* sistem

Proses evaluasi dilakukan dengan menggunakan parameter data yang dikirim oleh perangkat dan data yang diterima oleh Thingsboard, untuk mengetahui harak maksimal yang dapat dijangkau oleh ESP32 maka pengambilan data dilakukan sebanyak 3 kali yaitu pada jarak 10 meter, 20 meter, dan 25 meter. Untuk menghitung persentase *data loss* yang terjadi, persamaan yang digunakan adalah sebagai berikut:

$$Data\ loss = 100 - \left(\frac{JD}{JK} \times 100\right) \quad (1)$$

Keterangan:

JD = Jumlah data yang diterima
JK = Jumlah data yang dikirim

Real Time Operating System (RTOS) adalah sebuah sistem operasi yang dikembangkan untuk aplikasi *real time embedded system* yang berkembang di sekitar prosesor atau *controller*. Sebelum

ISSN: 1979-8415
E-ISSN: 2714-8025

adanya RTOS, *embedded system developer* menggunakan *primitive interrupt* untuk menjalankan proses multitasking. Akan tetapi, terdapat beberapa kelemahan pada sistem interrupt konvensional, terutama dalam penjadwalan task yang diberikan kepada embedded system (Jatmiko, 2004).

RTOS mengeksekusi program-program dalam sebuah pola yang teratur. RTOS tidak seperti sistem operasi yang pada umumnya digunakan pada komputer. Sistem operasi pada komputer akan bekerja sesaat ketika power masuk ke komputer, kemudian program komputer dijalankan. Sedangkan pada RTOS, sistem operasi ini dijalankan oleh sebuah program otomatis. Program tersebut merupakan sebuah kernel.

Ketika sistem dinyalakan, maka kernel akan menyala terlebih dahulu kemudian menyalakan *Real Time Operating Sistem*. Tugas utama dari RTOS adalah mengatur sumber daya yang ada meliputi prosesor, memori/register, serta hak akses menuju dua sumber daya terebut. Selain itu, RTOS bertugas melakukan komunikasi dan sinkronisasi. Selain berjalan menggunakan kernel RTOS dapat terdiri dari kombinasi beberapa *module kernel*, *file system*, *networking protocol stack*, dan komponen lain.

Kernel RTOS terdiri dari kernel object. Kernel object adalah blok-blok yang digunakan untuk membangun real time embedded system. RTOS kernel object tersebut yaitu:

1. Task

Task merupakan thread dari suatu proses yang akan dieksekusi oleh prosesor. Task bersifat konkuren dan independen terhadap task lain. Desain proses untuk real time application yaitu dengan membagi pekerjaan yang akan di kerjakan menjadi beberapa task (Chandane, 2016). Setiap task diberikan prioritas. Setiap task biasanya merupakan infinite loop dan terdiri dari 4 keadaan yaitu:

a. Running State

Running state merupakan keadaan dimana Task sedang berjalan atau dieksekusi. Ketika berada pada keadaan ini suatu task memanfaatkan prosesor dan sumber daya lain seperti register dan memori. Jumlah task yang dapat berada dalam keadaan ini sesuai dengan jumlah prosesor yang dimiliki oleh komputer atau mikrokontroller. Ketika berada pada keadaan running state, task dapat kembali ke ready state apabila muncul task baru yang memiliki prioritas yang lebih tinggi. Pada keadaan ini task dapat pula berpindah ke suspended state atau blocked state apabila melakukan hal berikut:

- Meminta resource yang tidak tersedia di dalam sistem.
- Meminta menunggu event tertentu muncul.
- Meminta delay.

b. Ready State

Sebuah task akan berada pada ready state setelah task tersebut dibuat. Ready state adalah keadaan dimana sebuah task siap untuk dieksekusi, task tersebut akan menunggu task lain yang memiliki prioritas sama atau lebih tinggi yang sedang dieksekusi pada running state. Pada keadaan ini task dapat berpindah ke running state atau suspended state.

c. Blocked state

Task akan berada pada blocked state apabila task tersebut sedang menunggu event lain terjadi. Event tersebut meliputi temporal event atau external event. Temporal event adalah delay yang diminta oleh task itu sendiri. Task akan di-unblock dijalankan kembali setelah periode delay habis. Sedangkan external event berupa queue atau semaphore event. Real time system membutuhkan keadaan ini karena jika state ini tidak ada maka task dengan prioritas rendah tidak memiliki kesempatan untuk dieksekusi atau dikenal dengan starvation.

d. Suspended State

Suspended state adalah keadaan dimana sebuah task ditunda eksekusinya untuk waktu yang tidak ditentukan. Penundaan tersebut dapat terjadi ketika task menunggu resources untuk dikosongkan terlebih dahulu. Task akan berada di keadaan ini apabila dipanggil fungsi vTaskSuspend(), dan akan keluar dari keadaan ini ketika dipanggil vTaskResume().

2. Semaphore

Semaphore merupakan objek berbentuk token yang nilainya dapat ditambah dan dikurangi oleh task untuk keperluan sinkronisasi ataupun mutual exclusion. Mutual exclusion adalah kondisi dimana terdapat sumber daya yang tidak dapat dipakai bersama dalam waktu yang bersamaan. Jadi, mutual exclusion terjadi ketika hanya ada satu proses yang boleh menggunakan sumber daya, dan proses lain yang ingin menggunakan sumber daya tersebut harus menunggu hingga sumber

ISSN: 1979-8415
E-ISSN: 2714-8025

daya tadi dilepaskan atau tidak ada proses yang menggunakan sumber daya tersebut.

Semaphore digunakan sebagai penanda bahwa task yang mengakuisisinya memiliki hak penuh untuk menjalankan operasi tertentu dan juga menggunakan recources yang ada. Akuisisi semaphore dibatasi oleh waktu dalam periode tertentu. Semaphore dapat diibaratkan sebagai kunci duplikat rumah yang dipegang oleh kepala keluarga. Anggota keluarga lain dapat meminjam kunci duplikat tersebut. Akan tetapi, kunci duplikat tersebut tersedia dalam jumlah terbatas sehingga tidak bisa dipinjam oleh sekaligus secara bersamaan (Jatmiko, 2004):

- Message Queues, merupakan struktur data yang digunakan untuk sinkronisasi, mutual exclusion, dan pertukaran data dengan mengantarkan pesan antar task. Keunggulan dari Real Time Operating System yaitu:
- Dapat berjalan pada perangkat keras dengan sumber daya yang terbatas.
- Memiliki tingkat modularitas yang tinggi.
- Dapat dikembangkan dan digunakan ulang.
- Memiliki beberapa fitur yang terdokumentasi lengkap
- Memiliki sistem pengaturan interrupt untuk proses-proses penting.
- Konfigurasi kernel dapat diatur secara detail oleh developer.

RTOS memiliki sejumlah karakteristik diantaranya reliability, predictable, performance, compactness, dan scalability.

1. Reliability

Reliability adalah keandalan atau konsistensi sistem dalam menjalankan tugasnya. Embedded system biasanya dibuat untuk bekerja dalam waktu yang lama tanpa adanya intervensi manusia. Maka dari itu embedded system harus bersifat reliable.

2. Predictability

RTOS menjamin setiap *real time system* dapat mengoperasikan tugasnya dalam waktu yang sudah ditentukan. Jadi perilaku RTOS itu sendiri dapat diprediksi atau dengan kata lain system call pada RTOS akan muncul dalam rentang waktu yang diketahui.

3. Performance

Pada umumnya embedded system harus dapat beroperasi dalam waktu yang cepat. Jumlah proses yang harus dikerjakan srta waktu antar proses yang singkat memberikan dampak pada kebutuhan CPU yang cepat. Performa dari prosesor dapat dideskripsikan dalam million instruction per second (MIPS). Performa tersebut dapat dipengaruhi oleh hardware ataupun software yang digunakan.

Selain mengukur MIPS, performa sistem juga dapat diukur dengan menggunakan throughput. Throughput adalaha rata-rata sebuah sistem dapat menjalankan keluaran dari masukan yang diterima. Definisi lainnya adalah bersaran yang menunjukkan ukuran data yang dapat dialirkan berdasarkan satuan waktu. Performa RTOS juga dapat diukur dengan menampilkan timestamp ketika sistem dijalankan sampai sistem selesai menjalankan tugasnya (Jatmiko, 2004).

1. *Routing*

Routing merupakan suatu protokol yang digunakan untuk mendapatkan rute dari suatu jaringan ke jaringan yang lain. Agar router dapat mengetahui bagaimana meneruskan paket-paket yang dikirim ke alamat yang dituju dengan menggunakan jalur terbaik, router menggunakan peta routing atau routing table.

a. *Static routing*, adalah salah satu jenis routing yang dimana pembuatan dan pembaharuan routing table dilakukan secara manual. Static routing tidak akan mengubah informasi yang ada pada table routing secara otomatis, sehingga admin harus mengubahnya secara manual apabila topologi jaringan berubah. Beberapa kelebihan dari static routing yaitu:

- Pemeliharaan bandwidth jaringan, karena pembaharuan informasi router membutuhkan broadcast tersur menerus.
- Keamanan jaringan, karena static routing hanya mengandung informasi yang telah dimasukkan secara manual.
- Sedangkan kekurangan dari static routing yaitu:
- Tidak adanya toleransi kesalahan, jika suatu router down maka static routing tidak akan memperbaharui informasi dan tidak akan menginformasikan ke router yang lainnya
- Pengembangan jaringan, jika suatu jaringan ditambah atau dipindahkan maka harus diperbaharui oleh administrator.

b. *Dynamic routing*, adalah jenis routing yang dimana pembuatan jalur dilakukan secara otomatis oleh router itu sendiri sesuai dengan konfigurasi yang dibuat. Jika ada perubahan topologi antar

ISSN: 1979-8415
E-ISSN: 2714-8025

jaringan, router secara otomatis akan membuat routing baru. Kelebihan dari dynamic routing adalah:

- Cocok untuk area yang luas.
- Hanya mengenalkan alamat yang terhubung langsung dengan routernya.
- Bila terjadi penambahan suatu jaringan maka tidak perlu semua router dikonfigurasi, hanya router yang berkaitan saja.
- Router berbagi informasi secara otomatis.
- Routing table dibuat secara dinamik.
- Tidak memerlukan campur tangan administrator.

Sedangkan kelemahan dynamic routing yaitu:

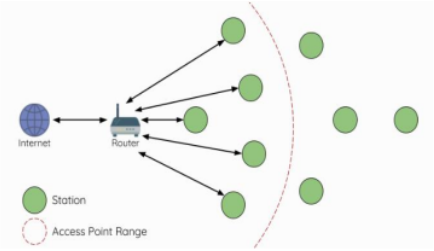
- Beban kerja router menjadi lebih berat.
- Kecepatan pengenalan dan kelengkapan routing table terbilang lama karena router mem-broadcast ke semua router lainnya sampai ada yang cocok sehingga setelah konfigurasi harus menunggu beberapa saat agar setiap router mendapatkan semua alamat yang ada.

2. *Esp Mesh*

Topologi mesh merupakan suatu bentuk dari topologi jaringan yang menghubungkan antara satu node dengan node lainnya didalam jaringan tersebut. Hubungan antar node ini akan membentuk sebuah rangkaian yang menyerupai jaring. Perbedaan utama pada topologi mesh dan topologi star adalah pada topologi mesh memiliki kemampuan untuk self-organizing dan self-configuration (Liu, 2017).

Topologi ini tidak membutuhkan dedicated link (perantara), setiap node akan saling berhubungan satu sama lainnya sehingga pada waktu yang sama akan terbentuk komunikasi langsung. Dikarenakan komunikasinya terhubung secara langsung, maka apabila ada satu node yang mati, maka tidak akan berpengaruh pada node yang lainnya.

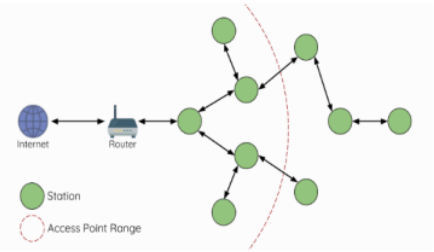
Pada arsitektur jaringan WiFi tradisional menggunakan point to multipoint dimana terdapat sebuah node pusat yang dikenal dengan Access Point (AP) yang terhubung langsung ke semua node (station) yang ada pada jaringan. Access point bertanggung jawab sebagai penengah dan meneruskan transmisi data antar station. Pada arsitektur tradisional ini memiliki kekurangan dimana area jangkauan yang kurang luas karena setiap station harus terhubung langsung dengan access point. Selain itu, jaringan ini rentan terjadi overloading dikarenakan jumlah station yang dapat terhubung ke access point terbatas. Gambar 5 menampilkan arsitektur jaringan WiFi tradisional.



Gambar 5. Arsitektur jaringan WiFi tradisional

Esp Mesh adalah sebuah protokol jaringan pada protokol WiFi. Esp Mesh memungkinkan banyak perangkat atau node tersebar di area yang luas untuk saling terhubung di suatu jaringan WLAN (Wireless Local Area Network). Esp Mesh dapat melakukan self-organizing dan self-healing, artinya jaringan ini dapat dibuat dan maintenance secara mandiri.

Esp Mesh berbeda dengan jaringan WiFi tradisional dimana setiap node tidak perlu untuk terhubung langsung ke node pusat. Sebagai gantinya, node diizinkan untuk terhubung dengan node tetangganya. Setiap node saling bertanggung jawab untuk menyampaikan transmisi data satu sama lain. Hal ini memungkinkan jaringan Esp Mesh untuk memiliki jangkauan yang lebih luas karena node masih dapat melakukan interkonektivitas tanpa perlu berada dalam jangkauan node pusat. Jaringan ini juga tidak rentan terhadap overloading karena jumlah node yang diizinkan pada jaringan tidak lagi dibatasi oleh node pusat (Access Point). Gambar 6 menampilkan arsitektur jaringan Esp Mesh.



Gambar 6. Arsitektur jaringan Esp Mesh

ISSN: 1979-8415
E-ISSN: 2714-8025

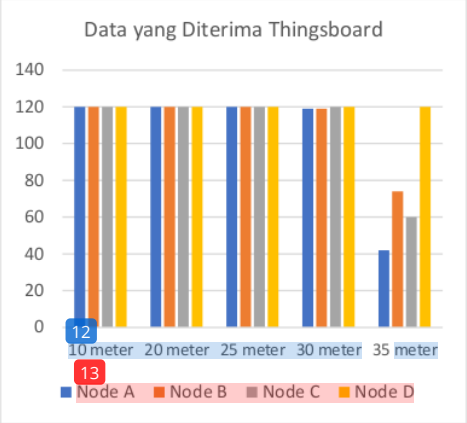
PEMBAHASAN

Pada bagian ini disajikan hasil pengujian kinerja sistem monitoring kualitas air tambak udang menggunakan real time operating system. Pengujian dilakukan selama 5 kali yaitu dengan jarak node sebesar 10 meter, 20 meter, 25 meter, 30 meter, dan 35 meter untuk mengetahui jarak paling optimal yang dapat dijangkau oleh ESP32. Ketika penguijan, baterai 18650 hanya mampu bertahan selama 3 jam. Setiap node mengirim data tiap menit sehingga dalam waktu 2 jam setiap node mengirim 120 data. Adapun jumlah data yang berhasil sampai ke Thingsboard dapat dilihat pada Tabel 1.

Tabel 1. Jumlah data yang sampai ke Thingsboard

Jarak Antar Node	Node A	Node B	Node C	Node D
10 meter	120	120	120	120
20 meter	120	120	120	120
25 meter	120	120	120	120
30 meter	119	119	120	120
35 meter	42	74	60	120

Berdasarkan Tabel 1, ketika jarak antar node yang digunakan sebesar 10 meter dari 480 data yang dikirim jumlah yang sampai ke server Thingsboard adalah 480 data, dimana data dari node A sebanyak 120 data, node B 120 data, node C 120 data, dan node D 120 data. Ketika jarak antar node yang digunakan sebesar 20 Meter, dari 480 data yang dikirim jumlah yang sampai ke server Thingsboard sebanyak 480 data, dimana data dari node A sebanyak 120 data, node B 120 data, node C 120 data, dan node D 120 data. Ketika jarak antar node yang digunakan sebesar 25 meter, dari 480 data yang dikirim data yang sampai ke Thingsboard sebanyak 480 data, dimana data dari node A sebanyak 120 data, node B 120 data, node C 120 data, dan node D 120 data. Ketika jarak antar node yang digunakan sebesar 30 meter, dari 478 data yang dikirim data yang sampai ke Thingsboard sebanyak 478 data, dimana data dari node A sebanyak 119 data, node B 119 data, node C 120 data, dan node D 120 data. Ketika jarak antar node yang digunakan sebesar 35 meter, dari 480 data yang dikirim data yang sampai ke Thingsboard sebanyak 296 data, dimana data dari node A sebanyak 42 data, node B 74 data, node C 60 data, dan node D 120 data.



Gambar 7. Grafik data yang diterima oleh Thingsboard

Berdasarkan Gambar 7, pada pengujian dengan jarak antar sensor 30 meter, terdapat 1 data dari node A dan juga 1 data dari node B yang hilang atau tidak sampai ke server Thingsboard. Kedua data tersebut menghilang pada saat pengiriman dari node B ke node D. Ketika pengujian dengan jarak antar sensor 35 meter, pada pengiriman data dari node A ke node B, terdapat 41 data yang hilang. Pada pengiriman data dari node B ke node D, terdapat 83 data yang hilang dimana terdiri dari 37 data dari node A dan 46 data dari node B itu sendiri. Sedangkan pada pengiriman data dari node C ke node D terdapat 60 data yang hilang. Pada jarak ini sudah tidak baik untuk digunakan karena jumlah *dataloss* yang terjadi cukup banyak. Hal ini dikarenakan koneksi setiap node pada jarak ini sering terputus sehingga memperbesar kemungkinan terjadinya *dataloss*.

Berdasarkan Gambar 7, semakin besar jarak antar node maka data yang diterima cenderung semakin berkurang. Hal ini terjadi karena semakin jauh jarak antar node maka sinyal yang ditangkap setiap node akan semakin lemah. Ada beberapa faktor yang dapat mempengaruhi pengiriman data, diantaranya:

1. Jarak. Jarak antara pemancar WiFi dalam hal ini access point dengan penerima (station) sangat menentukan kualitasnya. Semakin dekat jaraknya maka kekuatan sinyal yang diterima oleh station akan semakin kuat, dan sebaliknya jika keduanya berada pada jarak yang berjauhan maka sinyal akan semakin melemah.

ISSN: 1979-8415
E-ISSN: 2714-8025

2. Interferensi. Interferensi sering terjadi karena penggunaan channel frekuensi yang sama oleh dua bahkan lebih perangkat WiFi, sehingga hal tersebut dapat menyebabkan gangguan atau hambatan terhadap satu sama lainnya.
3. Kekuatan pemancar sinyal. Besarnya daya pancar yang dihasilkan dari antena akan sangat berpengaruh pada sisi penerimanya. Sejatinya kekuatan sinyal akan semakin berkurang ketika sampai di penerima. Faktor interferensi dan jarak adalah penyebab utamanya. Jadi semakin kuat sinyal yang dipancarkan oleh access point maka akan semakin besar sinyal yang diterima oleh penerima
4. Sensitivitas penerima. Kemampuan penerima juga akan mementukan kualitas sinyal yang diterima. Kemampuan itu ditenukan oleh batas minimal sinyal yang bisa diterima oleh penerima. Semakin kecil batas minimal sinyal yang bisa diterima makan semakin bagus kualitas sinyal yang ditangkap.
5. LOS (Line Of sight). LOS merupakan sebuah kondisi dimana pada area yang berupa garis lurus antara pemancar sinyal (access point) dan penerima tidak terhalang oleh suatu benda atau apapun. Kesempatan mendapat sinyal yang kuat bisa didapatkan jika penghalang antara kedua perangkat masih berukuran kecil atu bahkan tidak ada sama sekali. Jenis material yang menghalangi juga sangat menentukan
6. Cuaca. Keadaan cuaca buruk dapat berdampak terhadap kecepatan atau akses internet. Selain itu, kondisi cuaca yang buruk juga dapat menghambat frekuensi gelombang yang dipancarkan oleh modul WiFi. Hal ini disebabkan karena gelombang atau sinyal yang dipancarkan mendapat hambatan dari suatu materi dalam hal ini adalah butiran hujan. Saat hujan turun, gelombang yang dipancarkan oleh modul WiFi akan membentur butiran hujan tersebut sehingga akan memantul kearah lain. Akibatnya, kualitas sinyal yang diterima akan menurun dan gelombang yang seharusnya diterima menjadi kacau dan berbelok-belok. Selain pengaruh benturan dengan butiran hujan, kondisi angin kencang juga dapat mempengaruhi kekuatan sinyal WiFi.
7. Kelembaban. Kelembaban juga dapat mempengaruhi kekuatan sinyal WiFi. Akan tetapi tidak terlalu drastis sehingga dapat menyebabkan kegagalan penerimaan sinyal sama sekali. Kelembaban udara hanya dapat membiat sinyal lebih sulit dikirim secara efisien, sehingga menghasilkan kecepatan koneksi menjadi lebih lambat.

Tabel 2. Persentase *data loss*. Berdasarkan Tabel 2, tampak bahwa persentase data loss yang terjadi pada ESP32 yang menggunakan RTOS dapat mencapai 0% pada jarak 25 meter. Hal ini membuktikan bahwa dengan menggunakan RTOS, ESP32 dapat melakukan pengambilan data, transformasi data, routing, dan komunikasi dalam dalam selang waktu 1 menit dengan cukup baik.

Tabel 2. Persentase *data loss*

Jarak Antar Node	Node A	Node B	Node C	Node D	Rata-Rata
10 meter	0%	0%	0%	0%	0%
20 meter	0%	0%	0%	0%	0%
25 meter	0%	0%	0%	0%	0%
30 meter	0,8%	0,8%	0%	0%	0,4%
35 meter	65,0%	38,3%	50,0%	0%	38,3%

Dengan menggunakan Real Time Operating System, sistem yang dibuat menjadi lebih konsisten. Dapat dilihat dari cycle time dari node B pada Tabel 3. Cycle Time merupakan waktu yang dibutuhkan untuk menyelesaikan semua task atau waktu rata-rata penyelesaian dari task yang berulang. Berdasarkan pada tabel 3, sistem yang dibuat memiliki cycle time sekitar 60000 milidetik atau 1 menit sehingga pengiriman data yang dilakukan menjadi konsisten setiap 1 menit. Hal ini menjadi salah satu kelebihan dari Real Time Operating System yang dimana dapat membuat kinerja sebuah sistem menjadi lebih konsisten. Pada RTOS, terdapat API vTaskDelayUntil() yang berfungsi untuk memastikan frekuensi eksekusi task menjadi konstan. Berbeda dengan vTaskDelay() yang mana akan menentukan waktu task akan di-unblock relatif terhadap waktu pada saat vTaskDelay() dipanggil, sedangkan vTaskDelayUntil() akan menentukan waktu absolut kapan task akan di-unblock.

ISSN: 1979-8415
E-ISSN: 2714-8025

Tabel 1. *Cycle Time*

Timestamp	Cycle Time (ms)
2020-11-02 07:48:06	60000
2020-11-02 07:49:06	60001
2020-11-02 07:50:06	59999
2020-11-02 07:51:06	60000
2020-11-02 07:52:06	60000
2020-11-02 07:53:06	60000
2020-11-02 07:54:06	60000
2020-11-02 07:55:06	60001
2020-11-02 07:56:06	59999
2020-11-02 07:57:06	60001
2020-11-02 07:58:06	60000
2020-11-02 07:59:06	59999
2020-11-02 08:00:06	60000
2020-11-02 08:01:06	60001
2020-11-02 08:02:06	59999
2020-11-02 08:03:06	60000
2020-11-02 08:04:06	60001
2020-11-02 08:05:06	59999
2020-11-02 08:06:06	60000
2020-11-02 08:07:06	60000
2020-11-02 08:08:06	60000
2020-11-02 08:09:06	60001
2020-11-02 08:10:06	59999
2020-11-02 08:11:06	60001
2020-11-02 08:12:06	59999
2020-11-02 08:13:06	60000
2020-11-02 08:14:06	60000
2020-11-02 08:15:06	60000
2020-11-02 08:16:06	60000
2020-11-02 08:17:06	60000

KESIMPULAN

Dari hasil analisis yang telah dilakukan dalam pengujian sistem monitoring kualitas tambak udang menggunakan RTOS, dapat ditarik kesimpulan sebagai berikut:

1. Sistem monitoring kualitas air tambak udang dibuat dengan menggunakan RTOS untuk melakukan multitasking. Metode ini menggabungkan antara round-robin scheduling dan preemptive priority-based scheduling. Task yang akan dieksekusi akan diberikan waktu berapa lama eksekusinya menggunakan time slice dan akan dieksekusi sesuai antrian, akan tetapi jika terdapat task lain yang memiliki prioritas lebih tinggi yang tiba-tiba muncul, maka task yang sedang berjalan akan di suspend dan task yang memiliki prioritas lebih tinggi tersebut akan dieksekusi terlebih dahulu.
2. Hasil pengujian pada jarak 10 meter, 20 meter, dan 25 meter semua data diterima oleh server sehingga persentase data loss-nya 0%, pada jarak 30 meter 478 data yang diterima dengan persentase data loss sebesar 0,4%, dan pada jarak 35 meter hanya 296 data yang diterima oleh server dengan persentase data loss sebesar 38,3%.
3. Dengan menggunakan Real Time Operating System, setiap node mengirim data secara konsisten setiap 1 menit.

DAFTAR PUSTAKA

Alimuddin, 2015, Sistem Pengendalian Kadar pH, Suhu, dan Level Air Pada Model Miniatur Tambak Udang, *Jurnal Electro Luceat (JEC)*, Vol. 1 No. 1 Juli 2015.

Chandane, M.P., 2016, Real Time Operating System: A Complete Overview, *International Journal of Electrical and Electronics Engineers*, Vol No 8 Issue 1 January-June 2016.

Espressif Systems (Shanghai) Co. Ltd., 2016, *ESP-MESH*, diakses dari: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/mesh.html.

Jatmiko, W., 2004, Teori & Aplikasi Realtime Operating System, Fakultas Ilmu Komputer Universitas Indonesia.

Liu, Y. dan Tong, K., 2017, Wireless Mesh Network in IoT Networks, *International Workshop on Electromagnetics: Application and Student Innovation Competition*.

BIODATA PENULIS

Sabtian Juliana, lahir di Pare-Pare tanggal 5 Juli 1997. Saat ini tercatat sebagai mahasiswa tingkat akhir pada Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Adnan, S.T., M.T., Ph.D, lahir di Makassar tanggal 26 April 1974, menyelesaikan pendidikan S1 bidang Teknik Elektro dari Universitas Hasanuddin tahun 1998, S2 bidang Teknik Komputer dari Institut Teknologi Bandung tahun 2001, dan S3 bidang Computer Sciences dari University of Tsukuba tahun 2012. Saat ini tercatat sebagai Dosen Tetap Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan jabatan akademik Lektor Kepala pada bidang minat Internet of Things and Parallel Computing.

Ir. Christoforus Yohannes, M.T., lahir di Makassar tanggal 16 Juli 1960, menyelesaikan pendidikan S1 bidang Teknik Elektro dari Universitas

ISSN: 1979-8415
E-ISSN: 2714-8025

Hasanuddin tahun 1986, dan S2 bidang Teknik Elektro dari Universitas Hasanuddin tahun 2002. Saat ini tercatat sebagai Dosen Tetap Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan jabatan akademik Lektor pada bidang minat Robotika, Internet of Things, Instrumentasi Elektronika dan Teknik Kendali.

.

bukti_3283-Article_Text-5113-1-10-20210129.pdf

ORIGINALITY REPORT

20%	17%	6%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.warriornux.com Internet Source	2%
2	jurnal.umk.ac.id Internet Source	1%
3	taufikroom.wordpress.com Internet Source	1%
4	docplayer.info Internet Source	1%
5	yusniaalfisyahrin.wordpress.com Internet Source	1%
6	sayappatah69.blogspot.com Internet Source	1%
7	Submitted to Universitas Brawijaya Student Paper	1%
8	repository.upnvj.ac.id Internet Source	1%
9	Submitted to Universitas Sebelas Maret Student Paper	1%

10	j-ptiik.ub.ac.id Internet Source	1 %
11	sudardjattanusukma.wordpress.com Internet Source	1 %
12	gjoerdetselv.com Internet Source	1 %
13	www.mdpi.com Internet Source	1 %
14	text-id.123dok.com Internet Source	1 %
15	blog.unnes.ac.id Internet Source	<1 %
16	eprints.umg.ac.id Internet Source	<1 %
17	Oksana Sologub. "Crystal structure of $R_3Pd_{25-x}B_{8-y}$, R = La, Ce", Journal of Physics Conference Series, 06/01/2009 Publication	<1 %
18	mhrdika.wordpress.com Internet Source	<1 %
19	id.123dok.com Internet Source	<1 %
20	Submitted to American Museum of Natural History Student Paper	<1 %

21	digilib.unhas.ac.id Internet Source	<1 %
22	Syarif Mohamad Rizannur. "ANALISIS PERSEBARAN KEBISINGAN DI AREA PT. PLN (PERSERO) WILAYAH KALIMANTAN BARAT KABUPATEN KUBU RAYA DENGAN METODE NOISE MAPPING", Jurnal Teknologi Lingkungan Lahan Basah, 2016 Publication	<1 %
23	embedded.stei.itb.ac.id Internet Source	<1 %
24	Submitted to Universitas Nasional Student Paper	<1 %
25	embeddednesia.com Internet Source	<1 %
26	Dona Wahyudi, M. Udin Harun Al Rasyid, Iwan Syarif. "Implementasi Algoritma Clustering untuk Efisiensi Energi di Wireless Sensor Network", INOVTEK Polbeng - Seri Informatika, 2019 Publication	<1 %
27	socj.telkomuniversity.ac.id Internet Source	<1 %
28	Submitted to UIN Sultan Syarif Kasim Riau Student Paper	<1 %

29	Submitted to Wilson High School Student Paper	<1 %
30	core.ac.uk Internet Source	<1 %
31	jyu.tipografiamec.it Internet Source	<1 %
32	F. Balarin, M. Chiodo, P. Giusto, H. Hsieh, A. Jurecska, L. Lavagno, A. Sangiovanni-Vincentelli, E.M. Sentovich, K. Suzuki. "Synthesis of software programs for embedded control applications", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1999 Publication	<1 %
33	blog-dosen.binadarma.ac.id Internet Source	<1 %
34	eprints.umm.ac.id Internet Source	<1 %
35	ijcs.stmikindonesia.ac.id Internet Source	<1 %
36	jurnal.ugm.ac.id Internet Source	<1 %
37	repository.its.ac.id Internet Source	<1 %

38	Erlania Erlania, I Nyoman Radiarta. "PERBEDAAN SIKLUS TANAM BUDIDAYA RUMPUT LAUT, Kappaphycus alvarezii, TERHADAP VARIABILITAS TINGKAT SERAPAN KARBON", Jurnal Riset Akuakultur, 2014 Publication	<1 %
39	Sepriandi Parningotan, Tri Mulyanto. "RANCANG BANGUN PROTOTIPE ALAT PENGHITUNG PRODUK SECARA OTOMATIS DENGAN KONSEP INTERNET OF THING (IOT) BERBASIS MIKROKONTROLLER (ARDUINO UNO)", Electro Luceat, 2020 Publication	<1 %
40	id.scribd.com Internet Source	<1 %
41	journal.unibos.ac.id Internet Source	<1 %
42	library.binus.ac.id Internet Source	<1 %
43	repository.usd.ac.id Internet Source	<1 %
44	staff.uny.ac.id Internet Source	<1 %
45	supergirlsheaven.blogspot.com Internet Source	<1 %
	www.slideshare.net	

46	Internet Source	<1%
47	Sudirman Melangi, Muhammad Asri, Stephan Adriansyah Hulukati. "Sistem Monitoring Informasi Kualitas dan Kekeruhan Air Tambak Berbasis Internet of Things", Jambura Journal of Electrical and Electronics Engineering, 2022 Publication	<1%
48	Saiful Azhari Muhammad, Haryono Haryono. "Design of Pond Water Temperature Monitoring Built Using NodeMCU ESP8266", Sinkron, 2022 Publication	<1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		