

Pengembangan Aplikasi Desktop Monitoring Kualitas Air Berbasis IoT untuk Pemantauan Parameter PH, Suhu, Salinitas dan TDS Secara *Real-Time*

Choirun Nisa*¹

Politeknik NSC Surabaya, Indonesia¹

choyunnisa@gmail.com

*Corresponding author :choyunnisa@gmail.com

Abstrak— Penelitian ini bertujuan mengembangkan aplikasi desktop monitoring kualitas air berbasis *Internet of Things* (IoT) yang mampu menampilkan data pH, suhu, salinitas, Total *Dissolved Solid* (TDS), serta posisi *node sensor* secara *real-time*. Aplikasi dikembangkan menggunakan C# dengan *framework* .NET (WinForms) dan terintegrasi dengan sistem sensor berbasis *Ardupilot Mega* melalui komunikasi serial menggunakan modul radio 3DR frekuensi 433 MHz. Metode penelitian menggunakan *Software Development Life Cycle* (SDLC) model waterfall yang meliputi analisis kebutuhan, perancangan, implementasi, dan pengujian. Arsitektur aplikasi terdiri atas lapisan penerimaan data, pemrosesan data, penyimpanan data, dan presentasi. Hasil *black box testing* menunjukkan seluruh fitur berjalan sesuai kebutuhan, termasuk koneksi serial, parsing data, visualisasi *real-time*, penyimpanan data, dan pengukuran *Quality of Service* (QoS). Aplikasi menyajikan data dalam bentuk indikator numerik, grafik linier, dan grafik historis yang diperbarui otomatis. Pengujian performa menghasilkan rata-rata waktu pembaruan data 286,8 ms, waktu respons GUI 118,5 ms, throughput 2.388,89 bps, dan packet loss 0%, sedangkan rata-rata delay sebesar 40,34 detik masih memerlukan optimasi. Fitur tambahan seperti timer monitoring, jam *real-time*, dan indikator jarak GPS mendukung pemantauan kualitas air secara efektif dan diharapkan berkontribusi pada pengelolaan sumber daya perairan yang berkelanjutan.

Abstract— This study aims to develop and implement an *Internet of Things* (IoT)-based desktop application for water quality monitoring capable of displaying *real-time* pH, temperature, salinity, Total *Dissolved Solids* (TDS), and sensor node location data. The application was developed using C# with the .NET WinForms framework and integrated with an *Ardupilot Mega*-based sensor system through serial communication using a 433 MHz 3DR radio module. The research adopted the *Waterfall Software Development Life Cycle* (SDLC), consisting of requirements analysis, system design, implementation, and testing. The application architecture comprises four main layers: data acquisition, data processing, data storage, and data presentation. *Black-box testing* confirmed that all functional requirements were successfully implemented, including serial communication, data parsing, *real-time* visualization, data logging, and *Quality of Service* (QoS) monitoring. The application presents monitoring data through numerical indicators, line charts, and historical graphs that update automatically. Performance evaluation showed an average data refresh time of 286.8 ms, a GUI response time of 118.5 ms, a throughput of 2,388.89 bps, and 0% packet loss, while the average transmission delay of 40.34 seconds indicates the need for further optimization. Additional features, including a monitoring session timer, *real-time* clock, and GPS distance indicator, enhance operational efficiency and support sustainable water quality monitoring.

Keywords— *Internet of Things*; water quality monitoring; C#; *real-time* monitoring; QoS.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-nc/4.0/) license.



1. Pendahuluan

Manajemen kelestarian perairan yang efektif dan efisien menjadi hal yang semakin krusial di era modern ini. Berbagai faktor seperti meningkatnya populasi penduduk, buruknya manajemen limbah industri, eksploitasi potensi perairan yang tidak memperhatikan kelestarian lingkungan, serta perubahan iklim global telah memberikan tekanan besar terhadap kualitas ekosistem perairan [1][2][3]. Perairan dangkal seperti sungai, danau, dan wilayah pesisir menjadi yang paling terdampak, mengingat kawasan ini merupakan sumber air utama sekaligus media budidaya yang vital bagi masyarakat sehari-hari [4][5]. Metode konvensional dalam pemantauan kualitas air, yaitu pengambilan sampel untuk diuji di laboratorium, memiliki keterbatasan yang signifikan dari sisi waktu dan biaya. Proses ini tidak dapat memberikan gambaran kondisi perairan secara langsung (*real-time*), sehingga respons terhadap pencemaran atau perubahan kualitas air seringkali terlambat [6]. Beberapa sistem monitoring berbasis

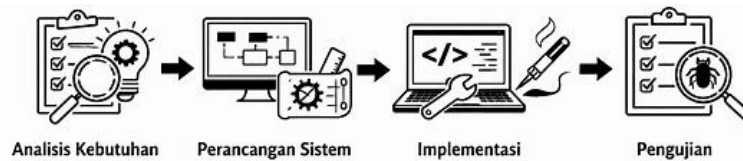
jaringan sensor nirkabel (*wireless sensor network*) yang telah ada sebelumnya pun masih menghadapi keterbatasan mendasar: infrastruktur komunikasi bersifat tetap (*fixed*) sehingga tidak fleksibel untuk dipindahkan antarwilayah perairan, tidak dilengkapi pelacakan posisi geografis (GPS) yang diperlukan untuk pemetaan spasial kualitas air, serta tidak menyediakan antarmuka visualisasi data secara *real-time* yang dapat diakses langsung oleh operator di lapangan [7].

Sebagai solusi atas permasalahan tersebut, penelitian-penelitian terkini mulai merancang sistem monitoring kualitas air terdistribusi yang memanfaatkan node sensor nirkabel (*wireless sensor node*) yang terhubung dalam satu jaringan terintegrasi, sehingga data dapat dikumpulkan dari berbagai titik perairan secara simultan dan otomatis [8]. Sistem ini memungkinkan pengukuran parameter fisika-kimia air seperti suhu pH, *Total Dissolved Solid* (TDS), salinitas, serta posisi geografis melalui GPS secara lebih efisien [9], [10]. Data yang dikumpulkan tidak hanya berguna untuk memantau kualitas air secara umum, tetapi juga untuk mendeteksi fenomena *upwelling* atau umbalan pada perairan budidaya ikan keramba jaring apung yaitu peristiwa naiknya massa air bawah yang kaya zat hara berlebih yang berpotensi mengakibatkan kematian ikan secara massal [11][12]. Namun demikian, tantangan berikutnya adalah bagaimana data yang dikumpulkan oleh sistem sensor tersebut dapat divisualisasikan secara mudah, cepat, dan akurat oleh pengguna. Tanpa adanya antarmuka pemantauan yang terintegrasi, data sensor yang berharga tidak dapat dimanfaatkan secara optimal untuk pengambilan keputusan [13]. Beberapa penelitian telah mengembangkan aplikasi berbasis *smartphone* maupun web untuk menampilkan data sensor secara *real-time* [14], [15], [16], namun masih banyak yang menggunakan protokol komunikasi yang berbeda antara sisi sensor dan sisi aplikasi, sehingga mengurangi efisiensi dan keandalan sistem secara keseluruhan [17]. Oleh karena itu, dibutuhkan sebuah aplikasi monitoring perairan berbasis perangkat lunak yang mampu menampilkan data hasil pembacaan sensor secara *real-time* dengan satu protokol komunikasi yang konsisten, sehingga operator, peneliti, maupun petani ikan dapat memantau kondisi perairan kapan saja dan di mana saja.

Meskipun berbagai penelitian sebelumnya telah mengembangkan sistem pemantauan kualitas air berbasis IoT, terdapat beberapa kelemahan mendasar yang belum terselesaikan secara memadai. Pertama, sebagian besar sistem yang ada mengandalkan aplikasi berbasis web atau *mobile* yang sangat bergantung pada konektivitas internet yang stabil, sehingga rentan terhadap gangguan layanan dan latensi tinggi pada area perairan terpencil [18][14], [19]. Kedua, banyak penelitian mengimplementasikan visualisasi data menggunakan protokol komunikasi yang heterogen antara lapisan sensor dan lapisan aplikasi, seperti kombinasi HTTP dan MQTT secara bersamaan, yang berpotensi menimbulkan *bottleneck* serta inkonsistensi data [20], [21]. Ketiga, fitur deteksi fenomena *upwelling* secara otomatis berbasis ambang batas parameter multi-sensor belum banyak diintegrasikan langsung ke dalam antarmuka pemantauan [22], [23]. Penelitian ini hadir untuk menjawab ketiga celah tersebut melalui tiga kontribusi utama: (1) pengembangan aplikasi monitoring berbasis desktop yang beroperasi secara mandiri tanpa ketergantungan penuh pada layanan cloud, sehingga menjamin ketersediaan sistem di lingkungan dengan infrastruktur jaringan terbatas; (2) penerapan protokol komunikasi serial secara konsisten dari node sensor hingga lapisan visualisasi aplikasi desktop, yang meminimalkan overhead konversi data dan meningkatkan keandalan transmisi *real-time*; serta (3) integrasi mekanisme deteksi dini *upwelling* berbasis analisis multi-parameter langsung pada antarmuka pengguna. Pemilihan platform desktop dibanding web atau *mobile* secara khusus didasarkan pada kebutuhan operator sistem di lapangan yang memerlukan akses data persisten, kemampuan *logging* lokal tanpa batas kuota, serta antarmuka yang lebih kaya fitur untuk analisis visualisasi multi-panel secara bersamaan serta aspek-aspek yang sulit dioptimalkan pada platform *mobile* maupun browser web dengan keterbatasan sumber daya komputasi dan ruang layar [24][25].

2. Metodologi Penelitian

Penelitian ini menggunakan pendekatan *Software Development Life Cycle* (SDLC) model *waterfall* [26] yang terdiri dari empat tahapan utama seperti ditunjukkan pada Gambar 1. Model *waterfall* dipilih dalam penelitian ini karena kebutuhan sistem telah terdefinisi dengan jelas sejak awal, yaitu menerima, memproses, dan memvisualisasikan data sensor kualitas air secara *real-time*. Pendekatan ini lebih sesuai dibandingkan metode *agile* yang mensyaratkan keterlibatan pengguna secara iteratif, maupun *prototyping* yang cocok apabila kebutuhan belum pasti. Selain itu, model *waterfall* memberikan alur pengembangan yang terstruktur dan terdokumentasi dengan baik, sehingga memudahkan proses verifikasi dan validasi pada setiap tahapan sebelum berlanjut ke tahap berikutnya.



Gambar 1. Metode Penelitian

Analisis Kebutuhan Aplikasi

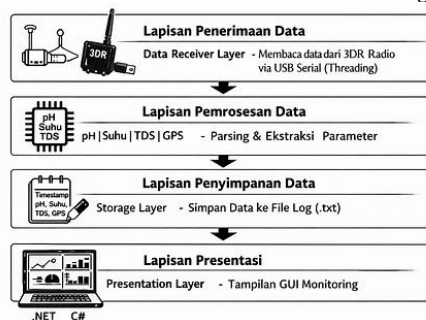
Pada tahap ini dilakukan identifikasi kebutuhan fungsional dan non-fungsional aplikasi berdasarkan hasil studi literatur dan analisis karakteristik data yang dikirimkan oleh platform perangkat keras. Kebutuhan fungsional aplikasi meliputi menerima *frame* data dari modul radio 3DR melalui koneksi serial USB secara kontinu, melakukan *parsing* data untuk mengekstrak nilai masing-masing parameter kualitas air dari *frame* data yang memiliki karakter *header* dan *terminator* khusus, menampilkan nilai parameter kualitas air (pH, suhu, TDS, salinitas) secara numerik dan grafis *real-time*, menampilkan grafik garis historis dengan sumbu-y sebagai besaran waktu dan sumbu-x sebagai nilai parameter kualitas air, menyimpan seluruh rekaman data ke dalam file teks (.txt) sebagai basis data local, kebutuhan non-fungsional meliputi kecepatan respons tampilan (*real-time performance*), kemudahan penggunaan antarmuka (*usability*), dan keandalan sistem dalam menerima data secara kontinu.

Perancangan Sistem

Gambar 2 menampilkan skema sistem visualisasi data monitoring perairan menggunakan modul Radio 3DR 433 MHz di sisi kiri sebagai penerima data dari node sensor yang terhubung melalui kabel USB Serial ke sebuah laptop di sisi kanan. Laptop tersebut menampilkan aplikasi GUI monitoring dengan grafik suhu, PH, TDS, salinitas, serta posisi GPS. Dari sistem tersebut terdapat 4 lapisan arsitektur utama yaitu lapisan penerima data, lapisan pemroses data, lapisan penyimpanan data dan lapisan presentasi seperti ditunjukkan pada Gambar 3.



Gambar 2. Skema Sistem Visualisasi Data Monitoring Perairan



Gambar 3. Lapisan arsitektur sistem

Berdasarkan Gambar 3 aplikasi disusun berdasarkan struktur lapisan sebagai berikut:

Lapisan Penerimaan Data (Data Receiver Layer)

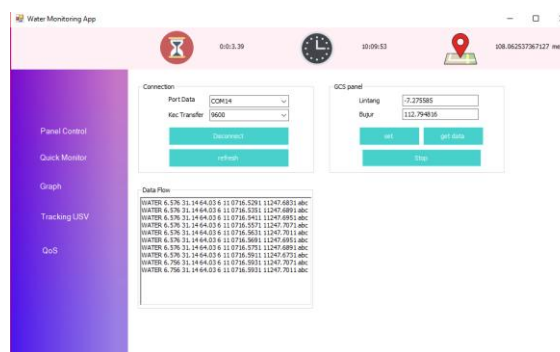
Lapisan penerimaan data bertugas membaca data yang masuk dari modul radio 3DR melalui port serial USB secara kontinu. Proses pembacaan dijalankan menggunakan mekanisme *threading* agar tidak mengganggu kinerja antarmuka pengguna. Alur proses sistem penerimaan dan *parsing* data sesuai Gambar 4 adalah aplikasi membuka koneksi serial pada port USB yang terhubung ke modul radio 3DR dengan konfigurasi *baud rate* yang sesuai. Pada penelitian ini digunakan *baud rate* 9600. Kemudian data yang masuk dibaca secara kontinu oleh *thread* penerimaan data di latar belakang (*background thread*). Lapisan pemrosesan data bertugas melakukan *parsing* terhadap *frame* data yang diterima. Proses *parsing* dilakukan dengan mengidentifikasi karakter *header* sebagai penanda awal *frame* dan karakter *terminator* sebagai penanda akhir *frame*, kemudian mengekstrak nilai masing-masing parameter kualitas air (pH, suhu, TDS, salinitas, latitude, longitude) dari dalam *frame* tersebut. Lapisan penyimpanan data bertugas menyimpan

seluruh data hasil *parsing* ke dalam file teks (.txt) secara berkala. Setiap baris rekaman mencakup *timestamp*, nilai pH, suhu, TDS, salinitas, latitude, dan longitude. Lapisan presentasi bertugas menampilkan data kepada pengguna melalui antarmuka grafis (GUI) yang dibangun menggunakan C# dengan framework .NET (WinForms/WPF). Implementasi aplikasi monitoring dilakukan menggunakan bahasa pemrograman C# dengan framework .NET (WinForms) pada lingkungan pengembangan Microsoft Visual Studio. Implementasi mencakup seluruh modul yang telah dirancang meliputi modul penerimaan data serial, modul parsing data, modul penyimpanan data, dan modul visualisasi grafis.

Pengujian dilakukan untuk memastikan aplikasi memenuhi seluruh kebutuhan yang telah ditetapkan. Pada penelitian ini digunakan pengujian *black box testing* untuk menguji apakah seluruh fitur aplikasi berjalan sesuai kebutuhan fungsional yang telah ditetapkan, meliputi penerimaan data serial, *parsing frame* data, tampilan numerik dan grafis *real-time*, tampilan peta serta penyimpanan data ke file .txt. Selain pengujian fungsional, dilakukan pula pengujian performa aplikasi untuk mengukur aspek kuantitatif sistem secara *real-time*, meliputi rata-rata waktu pembaruan data (*update rate*), jumlah data yang berhasil diproses per detik, serta waktu respons antarmuka pengguna (*GUI response time*). Kemudian terdapat pula pengujian kualitas komunikasi data yang ditampilkan pada menu QoS yaitu *delay* dan *packet loss*.

3. Hasil dan Pembahasan

Pada bagian ini dipaparkan hasil implementasi dan pengujian aplikasi monitoring kualitas perairan yang telah dikembangkan. Aplikasi masih memiliki beberapa keterbatasan, yaitu hanya berjalan pada sistem operasi Windows karena menggunakan framework .NET (WinForms), komunikasi data masih menggunakan modul radio 3DR frekuensi 433 MHz melalui koneksi serial USB, penyimpanan data masih bersifat lokal dalam format .txt, serta fitur Tracking USV masih dalam tahap pengembangan. Keterbatasan tersebut menjadi dasar pengembangan selanjutnya, seperti migrasi ke framework lintas platform, integrasi penyimpanan berbasis cloud, dan penyempurnaan fitur pelacakan USV.

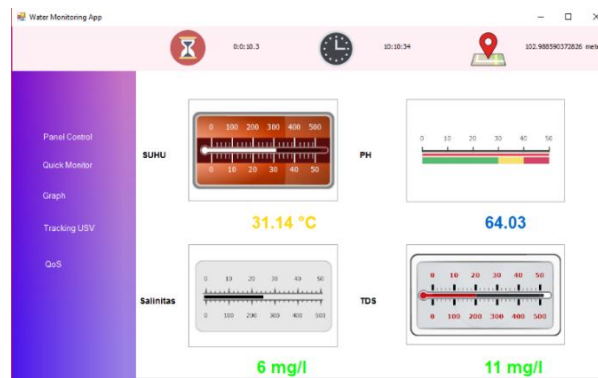


Gambar 4. Menu Panel Control

Gambar 4 menunjukkan tampilan Menu Panel Control yang terdiri atas Panel Connection, GCS Panel, dan Data Flow. Panel Connection digunakan untuk memilih port serial dan baud rate, GCS Panel berfungsi mengambil data GPS, sedangkan Data Flow menampilkan status penerimaan paket data. Pada menu ini diterapkan proses akuisisi data, yaitu memverifikasi frame dengan header "WATER" dan panjang minimal 49 byte sebelum dilakukan parsing, ditampilkan pada antarmuka, serta disimpan ke file .txt dan basis data.

3.1 Menu Panel Control

Gambar 4 menunjukkan tampilan dari menu *panel control*. Pada menu ini terdapat 3 bagian utama yaitu *panel connection*, *GCS panel* dan *data flow*. *Panel connection* digunakan untuk memilih port yang sesuai dengan port radio yang terhubung dan untuk memilih *baudrate* yang digunakan. *GCS panel* adalah panel yang berfungsi untuk mengambil data GPS. Sementara *data flow* berfungsi sebagai informasi adanya paket data yang masuk. Pada bagian bagian ini pula algoritma akuisisi data diterapkan pada sistem. Apabila data yang masuk memiliki header "WATER" dan memiliki panjang data lebih dari sama dengan 49 byte (jumlah byte minimum data dalam satu *frame* penuh tanpa *loss*) maka data tersebut akan diolah, *diparsing* untuk ditampilkan sesuai masing-masing nilai sensor dan disimpan pada .txt dan database.



Gambar 5. Menu Quick Monitor

Gambar 5 menampilkan antarmuka monitoring yang terdiri atas Quick Monitor, Graph, QoS, dan elemen informasi tambahan. Quick Monitor menyajikan nilai suhu, pH, salinitas, dan TDS dalam bentuk indikator numerik dan skala linier yang diperbarui secara real-time. Menu Graph menampilkan perubahan keempat parameter tersebut dalam bentuk grafik historis sehingga memudahkan operator memantau tren kualitas air. Hasil pengujian menunjukkan grafik relatif stabil dengan fluktuasi kecil yang masih berada pada batas normal, menandakan proses akuisisi, komunikasi serial, dan parsing data berjalan dengan baik. Menu QoS menampilkan parameter packet loss, throughput, dan delay untuk mengevaluasi performa komunikasi secara real-time. Selain itu, aplikasi dilengkapi dengan indikator durasi monitoring, jam real-time sebagai timestamp, serta informasi jarak antara server dan node sensor berdasarkan data GPS. Secara keseluruhan, seluruh fitur berfungsi sesuai rancangan dan mampu mendukung proses monitoring kualitas air secara real-time, sedangkan menu Tracking USV masih dalam tahap pengembangan.

Black Box Testing

Pengujian sistem dilakukan untuk memastikan seluruh fitur aplikasi monitoring berjalan sesuai dengan kebutuhan fungsional yang telah ditetapkan. Pengujian yang dilakukan meliputi pengujian fungsional (*black box testing*) terhadap seluruh fitur aplikasi. Hasil pengujian sistem ditunjukkan pada Tabel 1.

Tabel 1. Hasil Pengujian *Black Box Testing*

No	Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Koneksi Serial	Memilih port COM yang sesuai dan menekan tombol <i>Connect</i>	Aplikasi terhubung ke modul radio 3DR dan data mulai masuk pada <i>data flow</i>	Koneksi berhasil dan data raw tampil pada panel <i>data flow</i>	Berhasil
2	Deteksi Port Otomatis	Menekan tombol <i>Refresh</i> pada panel <i>connection</i>	Daftar port COM yang tersedia terdeteksi dan ditampilkan pada <i>ComboBox</i>	Seluruh port COM yang terhubung berhasil terdeteksi secara otomatis	Berhasil
3	Algoritma Akuisisi Data	Menerima frame data dengan header "WATER" dan panjang 49-53 byte	Hanya data yang memenuhi syarat yang diproses dan ditampilkan	Data valid berhasil diproses, data tidak valid dihitung sebagai paket <i>drop</i>	Berhasil
4	Parsing Data	Menerima frame data lengkap dari modul radio 3DR	Nilai suhu, pH, TDS, salinitas, latitude, dan longitude berhasil diekstrak dari frame data	Seluruh parameter berhasil di- <i>parsing</i> dan ditampilkan sesuai nilai masing-masing	Berhasil
5	Tampilan Numerik Real-Time	Data sensor diterima secara kontinu	Nilai parameter suhu, pH, salinitas, dan TDS diperbarui secara <i>real-time</i> pada label numerik	Nilai sensor berhasil ditampilkan dan diperbarui secara otomatis setiap frame data valid diterima	Berhasil
6	Tampilan Indikator Grafis (Quick Monitor)	Data sensor diterima secara kontinu	Indikator skala linier (<i>LinearScaleComponent</i>) bergerak sesuai nilai sensor secara <i>real-time</i>	Indikator grafis berhasil bergerak dan mencerminkan nilai sensor terkini secara akurat	Berhasil
7	Tampilan Grafik Historis (Graph)	Data sensor diterima secara kontinu selama beberapa menit	Grafik garis setiap parameter diperbarui secara <i>real-time</i> terhadap domain waktu	Grafik garis suhu, pH, salinitas, dan TDS berhasil ditampilkan dan diperbarui secara <i>real-time</i>	Berhasil
8	Timer Sesi Monitoring	Koneksi serial dibuka dan	Timer berjalan otomatis sejak koneksi dibuka dan menampilkan durasi sesi dalam format hh:mm:ss	Timer berhasil berjalan dan menampilkan durasi sesi monitoring secara akurat	Berhasil

		monitoring dimulai			
9	Penyimpanan Data ke File .txt	Data sensor diterima dan diproses oleh aplikasi	Setiap frame data valid tersimpan secara otomatis ke file data.txt dengan format timestamp dan nilai sensor	Data berhasil tersimpan secara otomatis pada file .txt dengan format yang sesuai	Berhasil
10	Perhitungan Jarak GPS	Sistem sensor digerakkan pada jarak tertentu dari server	Nilai jarak antara server dan node sensor ditampilkan dalam satuan meter secara <i>real-time</i>	Nilai jarak berhasil dihitung dan ditampilkan secara akurat berdasarkan selisih koordinat GPS	Berhasil
11	Tampilan Jam Real-Time	Aplikasi berjalan	Jam sistem ditampilkan dan diperbarui setiap detik pada bagian atas antarmuka	Jam real-time berhasil ditampilkan dan diperbarui secara kontinu	Berhasil
12	Menu QoS - Packet loss	Sistem dijalankan selama beberapa waktu	Nilai <i>packet loss</i> dihitung dan ditampilkan dalam satuan persen (%) secara <i>real-time</i>	Nilai <i>packet loss</i> berhasil dihitung dan ditampilkan sesuai rumus yang telah ditetapkan	Berhasil
13	Menu QoS - Throughput	Sistem dijalankan selama beberapa waktu	Nilai throughput dihitung dan ditampilkan dalam satuan Bps secara <i>real-time</i>	Nilai throughput berhasil dihitung dan ditampilkan secara akurat	Berhasil
14	Menu QoS - Delay	Data diterima dari sistem sensor	Nilai <i>delay</i> dihitung berdasarkan selisih waktu kirim dan waktu terima data	Nilai <i>delay</i> berhasil dihitung dan ditampilkan dalam satuan detik	Berhasil
15	Disconnect Serial	Menekan tombol <i>Disconnect</i> saat koneksi aktif	Koneksi serial tertutup dan timer berhenti	Koneksi berhasil diputus dan aplikasi kembali ke kondisi awal	Berhasil

Performa Aplikasi

Pengujian performa aplikasi dilakukan untuk mengevaluasi kemampuan sistem dalam memproses, menampilkan, dan merespons data yang diterima dari perangkat IoT melalui komunikasi serial menggunakan modul telemetri 3DR. Pengujian ini bertujuan untuk mengetahui tingkat kinerja aplikasi desktop yang dikembangkan dalam mendukung proses monitoring kualitas air secara *real-time*. Parameter yang diukur meliputi waktu pembaruan data (*data update time*), throughput sistem (jumlah data yang diproses per detik), *delay* dalam penerimaan data, *packet loss* (kehilangan paket dalam proses pengiriman data) dan waktu respons antarmuka pengguna (*GUI response time*). Seluruh pengujian dilakukan pada aplikasi desktop berbasis C# dengan konfigurasi komunikasi serial pada baud rate 9600 bps, serta dilakukan sebanyak 10 kali percobaan untuk memperoleh nilai rata-rata dari setiap parameter pengujian.

Tabel 2. Hasil Pengukuran Waktu Pembaruan Data

Parameter	P 1	P 2	P 3	P 4	P 5	P6	P7	P8	P 9	P 10	Rata-rata
Waktu Pembaruan Data (ms)	275	292	281	301	295	287	278	294	289	276	286,8

Tabel 3. Hasil Pengukuran Throughput Sistem

Parameter	P 1	P 2	P 3	P 4	P 5	P6	P7	P8	P 9	P 10	Rata-rata
Throughput (bps)	2319,52	2259,60	2474,52	2410,32	2388,74	2441,58	2365,21	2394,65	2428,41	2406,32	2388,89

Tabel 4. Hasil Pengukuran Waktu Respons GUI

Parameter	P 1	P 2	P 3	P 4	P 5	P6	P7	P8	P 9	P 10	Rata-rata
Waktu Respons GUI (ms)	112	119	121	116	123	118	115	120	117	124	118,5

Tabel 5. Hasil Pengujian Delay

Parameter	P 1	P 2	P 3	P 4	P 5	P6	P7	P8	P 9	P 10	Rata-rata
Delay (detik)	40,28	39,84	41,15	40,76	39,91	40,53	40,12	39,67	40,84	40,31	40,34

Tabel 6. Hasil Pengujian Packet loss

Parameter	P 1	P 2	P 3	P 4	P 5	P6	P7	P8	P 9	P 10	Rata-rata
Total Paket Diterima	263	421	154	398	275	312	287	356	301	324	309,1
Total Paket Drop	0	0	0	0	0	0	0	0	0	0	0
Packet loss (%)	0	0	0	0	0	0	0	0	0	0	0

Pengujian performa aplikasi secara kuantitatif melalui sepuluh kali pengukuran (P1–P10) pada masing-masing parameter hasil pengukuran disajikan pada Tabel 2 hingga Tabel 6. Berdasarkan Tabel 2, rata-rata waktu pembaruan data (*update rate*) sebesar 286,8 ms. Nilai ini menunjukkan aplikasi mampu *refresh* tampilan sensor kurang dari 300 ms per siklus, yang tergolong responsif untuk kebutuhan pemantauan lingkungan perairan. Hasil ini sejalan dengan penelitian [7] yang menekankan bahwa sistem monitoring kualitas air berbasis IoT yang efisien harus mampu memberikan pembaruan data secara cepat agar operator dapat merespons perubahan kondisi perairan secara tepat waktu. Berdasarkan Tabel 3, throughput sistem rata-rata sebesar 2.388,89 bps.

Nilai ini mendekati batas teoretis kecepatan transmisi serial yang digunakan, yaitu *baud rate* 9600 bps, sehingga mengindikasikan efisiensi pemanfaatan kanal komunikasi yang cukup tinggi oleh mekanisme *background thread* yang diterapkan. Selain itu, berdasarkan Tabel 6, hasil pengujian *packet loss* menunjukkan bahwa dari total paket yang diterima pada setiap sesi pengujian (rata-rata 309,1 paket), tidak ditemukan satu pun paket yang gagal diterima secara utuh (*packet loss* sebesar 0%). Hal ini membuktikan keandalan tinggi dari protokol komunikasi serial melalui modul radio 3DR pada kondisi pengujian yang dilakukan, sejalan dengan karakteristik komunikasi serial yang bersifat point-to-point dan minim gangguan interferensi dibanding komunikasi nirkabel berbasis jaringan seluler atau Wi-Fi yang lebih rentan terhadap keterlambatan respons [6].

Akan tetapi, berdasarkan Tabel 5, rata-rata *delay* pengiriman data tercatat sebesar 40,34 detik, sebuah nilai yang relatif tinggi dan perlu mendapat perhatian. *Delay* ini diduga disebabkan oleh waktu yang dibutuhkan untuk memperoleh fix koordinat GPS yang valid pada modul sensor sebelum data dikirimkan, mengingat *delay* diukur berdasarkan selisih waktu antar pengiriman frame data yang menyertakan informasi posisi. Nilai *delay* yang relatif besar ini menunjukkan adanya *trade-off* antara keandalan data (*packet loss* 0%) dan kecepatan pengiriman, yang konsisten dengan temuan penelitian [18] yang menyoroti bahwa sistem monitoring berbasis jaringan nirkabel jarak jauh cenderung menghadapi *trade-off* antara keandalan transmisi data dan kecepatan pengiriman, terutama pada kondisi infrastruktur jaringan yang terbatas.

Berdasarkan Tabel 4, waktu respons GUI rata-rata sebesar 118,5 ms. Nilai ini berada jauh di bawah ambang batas 200 ms yang umum ditetapkan sebagai batas persepsi manusia terhadap respons sistem yang terasa instan, sehingga operator dapat memantau kondisi perairan tanpa jeda yang mengganggu meskipun *delay* pengiriman data dari sensor relatif tinggi. Keberhasilan ini didukung oleh arsitektur pemisahan *thread* antara penerimaan data serial dan proses *rendering* antarmuka, sehingga tidak terjadi pembekuan GUI (*UI freeze*). Secara keseluruhan, hasil pengujian performa ini mengonfirmasi bahwa sistem yang dikembangkan memiliki keandalan komunikasi data yang tinggi (*packet loss* 0%) dan responsivitas antarmuka yang baik, meskipun masih terdapat ruang perbaikan pada aspek *delay* pengiriman data yang dapat menjadi fokus pengembangan lanjutan, sebagaimana dibandingkan dengan sistem berbasis web maupun mobile yang memiliki ketergantungan pada infrastruktur jaringan [18][19].

4. Kesimpulan

Berdasarkan hasil penelitian, aplikasi desktop monitoring kualitas air berbasis IoT berhasil dikembangkan menggunakan C# dan .NET (WinForms) serta mampu menerima, memproses, dan memvisualisasikan data kualitas air secara *real-time* melalui komunikasi serial menggunakan modul radio 3DR. Hasil pengujian menunjukkan bahwa sistem memiliki rata-rata waktu pembaruan data sebesar 286,8 ms, *throughput* sebesar 2.388,89 bps, waktu respons GUI sebesar 118,5 ms, dan *packet loss* sebesar 0%, sehingga menunjukkan bahwa sistem memiliki kinerja dan keandalan komunikasi yang baik. Namun, rata-rata *delay* sebesar 40,34 detik masih menjadi aspek yang perlu ditingkatkan. Kontribusi utama penelitian ini terletak pada pengembangan aplikasi monitoring berbasis desktop yang dapat beroperasi secara mandiri tanpa ketergantungan penuh pada layanan *cloud*, sehingga tetap dapat digunakan pada lingkungan dengan keterbatasan infrastruktur jaringan. Selain itu, penelitian ini menerapkan komunikasi serial secara *end-to-end* dari node sensor hingga aplikasi desktop untuk meningkatkan keandalan transmisi data secara *real-time*, serta mengintegrasikan mekanisme deteksi dini *upwelling* berbasis analisis multi-parameter secara

langsung pada antarmuka pengguna. Meskipun sistem telah berfungsi dengan baik, penelitian ini masih memiliki beberapa keterbatasan, yaitu aplikasi hanya dapat dijalankan pada sistem operasi Windows, masih menggunakan komunikasi serial berbasis 3DR, serta fitur *tracking* USV secara penuh. Oleh karena itu, penelitian selanjutnya dapat difokuskan pengembangan aplikasi multiplatform, serta penyempurnaan fitur navigasi dan *tracking* USV secara otomatis untuk meningkatkan fleksibilitas dan cakupan pemantauan kualitas air.

Daftar Pustaka

- [1] S. Das, V. K. Parida, C. S. Tiwary, A. K. Gupta, and S. Chowdhury, "Emerging Contaminants in the Aquatic Environment: Fate, Occurrence, Impacts, and Toxicity," in *ACS Symposium Series*, vol. 1475, M. P. Shah, K. Vasantharaj, and N. Saranya, Eds., Washington, DC: American Chemical Society, 2024, pp. 1–32. doi: 10.1021/bk-2024-1475.ch001.
- [2] J. J. H. Rojas, R. F. A. Bonifacio, J. M. B. Peña, D. R. Q. Leon, C. D. R. Peña, and K. G. P. Torres, "Sensors and IoT for Water Quality Monitoring: A Systematic Review of Technologies and Field Validation," *Journal of Robotics and Control (JRC)*, vol. 6, no. 6, pp. 2844–2858, 2025, doi: 10.18196/jrc.v6i6.28457.
- [3] W. H. Sugiharto, A. Bardadi, and E. Alfonsius, "Penilaian Kualitas Air Secara Real-Time Menggunakan IoTWQI dan Internet of Things," *Jurnal Ilmiah Sistem Informasi Akuntansi*, vol. 5, no. 1, pp. 73–85, 2025, doi: 10.33365/jimasia.v5i1.489.
- [4] J. Edokpayi, J. Odiyo, and O. Durowoju, "Impact of Wastewater on Surface Water Quality in Developing Countries: A Case Study of South Africa," 2017, pp. 401–416. doi: 10.5772/66561.
- [5] F. Mohseni *et al.*, "Ocean water quality monitoring using remote sensing techniques: A review," *Mar. Environ. Res.*, vol. 180, p. 105701, 2022, doi: <https://doi.org/10.1016/j.marenvres.2022.105701>.
- [6] "Water Environment Monitoring System Based on Wireless Sensor Network," *Academic Journal of Environmental Biology*, vol. 2, no. 1, 2021, doi: 10.38007/AJEB.2021.020105.
- [7] A. Ghosh *et al.*, "A Cost-Effective Real-Time Water Quality Monitoring System," in *2024 IEEE International Conference on Power, Electrical, Electronics and Industrial Applications (PEEIACON)*, IEEE, 2024, pp. 631–636. doi: 10.1109/PEEIACON63629.2024.10800410.
- [8] M. Al Drees *et al.*, "IoT-Based System for Real-Time Water Quality Monitoring and Advanced Turbidity and pH Sensor Calibration to Improve Accuracy and Reliability Using ThingSpeak," *IoT*, vol. 7, no. 2, p. 42, 2026, doi: 10.3390/iot7020042.
- [9] M. Komarudin, H. D. Septama, T. Yulianti, A. Yudamson, J. Hendri, and M. A. D. Arafat, "Multi node sensors for water quality monitoring towards precision aquaculture," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 739, no. 1, p. 012026, 2021, doi: 10.1088/1755-1315/739/1/012026.
- [10] D. R. Thévenot, K. Toth, R. A. Durst, and G. S. Wilson, "Electrochemical biosensors: recommended definitions and classification," *Biosens. Bioelectron.*, vol. 16, no. 1–2, pp. 121–131, 2001, doi: 10.1016/S0956-5663(01)00115-4.
- [11] A. Zuhaer, A. Khandoker, N. Enayet, P. K. P. Partha, and Md. A. Awal, "Sustainable aquaculture: An IoT-integrated system for real-time water quality monitoring featuring advanced do and ammonia sensors," *Aquac. Eng.*, vol. 112, p. 102620, 2026, doi: 10.1016/j.aquaeng.2025.102620.
- [12] M. Faisal, A. N. Bachtiar, and M. Darwis, "IoT Implementation for Hydroponic Water Monitoring Using Web-Based pH and TDS Sensors with Node-Red," *JISA (Jurnal Informatika dan Sains)*, vol. 8, no. 1, pp. 81–91, 2025, doi: 10.31326/jisa.v8i1.2209.
- [13] S. Sathyamoorthy, U. O. Matthew, T. S. Adekunle, and N. U. Okafor, "Advances and Challenges in IoT Sensors Data Handling and Processing in Environmental Monitoring Networks," *HAFED POLY Journal of Science, Management and Technology*, vol. 5, no. 2, pp. 40–60, 2024, doi: 10.4314/hpjsmt.v5i2.3.
- [14] Z. Mohd Yusof, M. M. Billah, and K. Kadir, "Real-time water quality monitoring system: an IoT application," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 1, p. 178, 2019, doi: 10.11591/ijeecs.v15.i1.pp178-182.
- [15] N. Naj and A. Sanzgiri, "An IoT based Real-Time Monitoring of Water Quality System," *ASIAN JOURNAL OF CONVERGENCE IN TECHNOLOGY*, vol. 7, no. 2, pp. 44–51, 2021, doi: 10.33130/AJCT.2021v07i02.010.
- [16] J. E. Candra and H. Hazimah, "IoT-Based Smart Water Quality Monitoring System for Early Detection of Water Pollution in Batam City," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 7, no. 4, pp. 1226–1234, 2025, doi: 10.47709/cnahpc.v7i4.7115.
- [17] S. Thouti, N. Venu, D. R. Rinku, A. Arora, and N. Rajeswaran, "Investigation on identify the multiple issues in IoT devices using Convolutional Neural Network," *Measurement: Sensors*, vol. 24, p. 100509, 2022, doi: 10.1016/j.measen.2022.100509.
- [18] N. Abdulkareem, S. Zeebaree, M. M. Sadeeq, D. Ahmed, A. Sami, and R. Zebari, "IoT and Cloud Computing Issues, Challenges and Opportunities: A Review," *Qubahan Academic Journal*, vol. 1, pp. 1–7, Mar. 2021, doi: 10.48161/qaj.v1n2a36.

- [19] N. A. Mohd Jais, A. F. Abdullah, M. Kassim, M. Karim, M. Abdulsalam, and N. A. Muhadi, "Improved accuracy in IoT-Based water quality monitoring for aquaculture tanks using low-cost sensors: Asian seabass fish farming," *Heliyon*, vol. 10, p. e29022, Apr. 2024, doi: 10.1016/j.heliyon.2024.e29022.
- [20] F. Foukalas, V. Gazis, and N. Alonistioti, "Cross-layer design proposals for wireless mobile networks: A survey and taxonomy," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 1, pp. 70–85, 2008, doi: 10.1109/COMST.2008.4483671.
- [21] R. Morabito, J. Kjallman, and M. Komu, "Hypervisors vs. Lightweight Virtualization: A Performance Comparison," in *2015 IEEE International Conference on Cloud Engineering (IC2E)*, IEEE, 2015, pp. 386–393, doi: 10.1109/IC2E.2015.74.
- [22] M. Z. Ulhaq, M. Farid, Z. I. Aziza, T. M. F. Nuzullah, F. Syakir, and N. R. Sasmita, "Forecasting Upwelling Phenomena in Lake Laut Tawar: A Semi-Supervised Learning Approach," *Infolitika Journal of Data Science*, vol. 2, no. 2, pp. 53–61, 2024, doi: 10.60084/ijds.v2i2.211.
- [23] N. Stojanovic and S. Chaudhary, "Real-Time Water Quality Monitoring in Aquaculture using IoT Sensors and Cloud-Based Analytics," *Research Journal of Computer Systems and Engineering*, vol. 4, no. 2, pp. 174–187, 2023, doi: 10.52710/rjcse.86.
- [24] Z. Zhao and Q. Hu, "The Application of a Computer Monitoring System Using IoT Technology," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–11, Jun. 2022, doi: 10.1155/2022/4033886.
- [25] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.comnet.2010.05.010.
- [26] W. W. Royce, "Managing the Development of Large Software Systems (1970)," in *Ideas That Created the Future*, H. R. Lewis, Ed., The MIT Press, 2021, pp. 321–332. doi: 10.7551/mitpress/12274.003.0035.