

# Pengendalian Sistem Irigasi Berbasis Komunikasi Radio *Full Duplex* dengan Algoritma *Decision Tree*

Indra Hermawan, Asep Kurniawan\*, Maria Agustin, Dimas Aulia Fachrudin, Nisrina Tsany Sulthanah, Arif Setiawan

Teknik Multimedia dan Jaringan, Politeknik Negeri Jakarta  
Kota Depok, Jawa Barat

[indra.hermawan@tik.pnj.ac.id](mailto:indra.hermawan@tik.pnj.ac.id), [asep.kurniawan@tik.pnj.ac.id](mailto:asep.kurniawan@tik.pnj.ac.id), [maria.agustin@tik.pnj.ac.id](mailto:maria.agustin@tik.pnj.ac.id)

Diterima: 12 November 2022. Disetujui: 5 Maret 2023. Dipublikasikan: 16 Maret 2023.

**Abstract** - An irrigation canal technology that is able to flow through all agricultural land according to the water needs of each land is demanded. In this case, smart irrigation systems can be a solution to the said problem. The previously developed smart irrigation technology has several limitations, such as the length of time data transmission taken to reach the client, sensor devices in the field must take turns sending data and receiving data resulting in data not being sent or not being read properly. On the other side, the system that has been developed previously heavily dependent on one factor, which is the water level to determine the degree of which how much the floodgate will be opened. In this research, a fast sensor-to-cloud communication system is designed using the WebSocket technology, as well as using artificial intelligence, namely decision tree algorithm. This algorithm is used to regulate the openness of the floodgates based on the water requirement for the rice fields, the water level in the irrigation canals, and the speed of the water flow. This research resulted in a prototype product of a smart irrigation system that can transmit data to clients with an average time of 417ms, with the system also able to handle 500 clients that are connected directly to the server. The system can also send and receive data simultaneously with a difference in time when data is received at the sensor node or at the central node which can reach 0ms. The artificial intelligence that was created using the decision tree method with an entropy algorithm and a random splitter can provide 100% accuracy. At the same time, the artificial intelligence webservice can handle 500 concurrent requests with a response time of 544ms.

**Keywords:** artificial intelligence, decision tree, full duplex communication, smart irrigation, web socket

**Abstrak**—Kebutuhan akan teknologi saluran irigasi yang mampu mengaliri seluruh lahan pertanian sesuai dengan kebutuhan air masing-masing lahan. Dalam hal ini, sistem irigasi cerdas dapat menjadi solusi dari permasalahan tersebut. Teknologi smart irigasi yang dikembangkan sebelumnya memiliki beberapa keterbatasan seperti lamanya waktu pengiriman data yang dibutuhkan untuk sampai ke klien, perangkat sensor di lapangan harus bergiliran mengirim dan menerima data sehingga mengakibatkan data tidak terkirim atau tidak terbaca dengan baik. Di sisi lain, sistem yang telah dikembangkan sebelumnya sangat bergantung pada satu faktor yaitu tinggi muka air untuk menentukan seberapa besar pintu air akan dibuka. Pada penelitian ini dirancang sistem komunikasi cepat *sensor-to-cloud* dengan menggunakan teknologi *WebSocket*, serta menggunakan kecerdasan buatan yaitu Algoritma *Decision Tree*. Algoritma ini digunakan untuk mengatur bukaan pintu air berdasarkan kebutuhan air sawah, tinggi muka air di saluran irigasi, dan kecepatan aliran air. Penelitian ini menghasilkan produk prototipe sistem irigasi cerdas yang dapat mengirimkan data ke klien dengan waktu rata-rata 417ms, dengan sistem tersebut juga mampu menangani 500 klien yang terhubung langsung ke server. Sistem juga dapat mengirim dan menerima data secara bersamaan dengan perbedaan waktu saat data diterima di sensor node atau di *central node* yang bisa mencapai 0ms. Kecerdasan buatan yang dibuat menggunakan metode pohon keputusan dengan Algoritma *Entropy* dan *random splitter* dapat memberikan akurasi 100%. Pada saat yang sama, layanan web kecerdasan buatan dapat menangani 500 permintaan bersamaan dengan waktu respons 544ms.

**Kata kunci:** *decision tree*, irigasi cerdas, kecerdasan buatan, komunikasi *full duplex*, *websocket*

## I. PENDAHULUAN

Permintaan pangan meningkat secara signifikan, terutama di bidang pertanian, yang melibatkan pergeseran dari pertanian konvensional

ke pertanian modern [1]. Pertanian merupakan salah satu sektor penunjang perekonomian nasional Indonesia sebagaimana perannya dalam penyediaan pangan, terutama beras yang merupakan makanan pokok penduduk Indonesia. Namun, produk pangan

pertanian belum dapat memenuhi kebutuhan masyarakat Indonesia. Penyediaan, pengelolaan, dan pendistribusian air pada lahan pertanian merupakan bagian dari faktor penting yang mempengaruhi kuantitas dan kualitas hasil budidaya sektor pertanian. [2]. Tanaman padi membutuhkan air untuk setiap jenis padi dan juga tergantung pada volume air yang berbeda untuk setiap fase pertumbuhan. Dalam penelitian yang dilakukan oleh [3] mengolah lahan/sawah menjadi beberapa kategori diantaranya, kategori A dengan lahan yang selalu tergenang air, kategori B dengan lahan yang tergenang setiap 4 hari sekali, dan kategori C yaitu lahan yang tergenang setiap 8 hari sekali. Penelitian ini telah membuktikan dari beberapa percobaan bahwa sawah kategori B memiliki kondisi pengairan sawah yang paling baik karena ketersediaan air yang cukup. Pada percobaan kondisi A, rata-rata tinggi tanaman padi 141,33 cm, dan jumlah anakan 19,22 batang/rumpun. Pada kondisi B rata-rata tinggi padi 133,55 cm dan jumlah anakan 23,56 batang/rumpun. Sedangkan pada kondisi C dengan frekuensi penyiraman 9 hari sekali pada tanaman padi menghasilkan tanaman padi dengan tinggi rata-rata 132,67 cm dan jumlah anakan 22,79 batang/rumpun. Sebagai sistem drainase menurut Peraturan Menteri PUPR No.30/PRT/M/2015 disebutkan bahwa salah satu pilar irigasi adalah ketersediaan pasokan air. Oleh karena itu diperlukan suatu teknologi yang dapat meningkatkan produktivitas tanaman pertanian, seperti adanya teknologi saluran irigasi yang mampu mengalirkan seluruh lahan pertanian sesuai dengan kebutuhan air masing-masing lahan. [2].

Permasalahan yang sering terjadi pada sistem irigasi yang ada adalah sulitnya pendistribusian air pada pintu air sistem irigasi yang mengakibatkan kurangnya pasokan air pada musim kemarau dan kelebihan pasokan air pada musim hujan yang mengalir melalui lahan pertanian.[4]. Dalam penelitian [5] latar belakang masalah tersebut adalah distribusi air yang tidak merata pada lahan pertanian karena kepentingan individu. Berdasarkan [6], sawah skala besar masih mengandalkan pekerja jasa untuk menutup dan membuka pintu air di bendungan secara manual untuk memenuhi kebutuhan air sawah setiap hari.

Selain masalah ketersediaan air, sistem pemantauan dan pengendalian aliran irigasi dalam pelaksanaannya juga membutuhkan sistem komunikasi yang dapat bekerja secara *real time*. Hal ini harus dilakukan agar data yang terekam oleh sensor dapat ditampilkan dan diproses dalam waktu singkat. Sistem dapat dikatakan bekerja secara *real time* jika dapat menampilkan data sensor ke *user interface* dalam waktu kurang dari 300ms[7]. Sistem irigasi saat ini sedang dikembangkan seperti di [1]

[4] [8] mengandalkan teknologi http untuk transmisi data. Dalam penelitian [1] waktu yang dibutuhkan perangkat untuk mengirim data ke server adalah 1,29 detik. Penelitian [8] juga menggunakan teknologi http untuk mengirimkan data, sistem yang dikembangkan membutuhkan waktu 3 detik untuk mentransfer data. Hal yang sama juga terjadi dalam penelitian [4], dalam 1 kali percobaan, perangkat mengirimkan 16 data ke server atau untuk mengirimkan 1 data membutuhkan waktu 3,7 detik. Penelitian [9] [10] masih menggunakan teknologi GSM untuk mengirimkan data, komunikasi berbasis GSM akan memakan biaya lebih mahal dibandingkan komunikasi berbasis IP [4]. Teknologi GSM juga tidak dirancang untuk mengirimkan data dengan cepat dan mengirimkan data secara intens. Penelitian [10] hanya mengirimkan informasi dalam kondisi tertentu seperti ketika ketinggian air melebihi 30cm. Hal ini tentu saja tidak memenuhi standar komunikasi *real time*.

Sistem irigasi lahan pertanian yang dikembangkan sebelumnya pada umumnya menggunakan topologi *star* yang terhubung langsung dengan server menggunakan koneksi berbasis WIFI atau seluler. Penelitian [11][12][13][14] menggunakan topologi yang mengirimkan data langsung dari sensor ke server. Penelitian [11][14] menggunakan teknologi *ethernet shield* untuk berkomunikasi ke internet. Ini tidak praktis untuk diterapkan di area yang luas karena akan membutuhkan kabel yang sangat panjang. Sedangkan penelitian [12] menggunakan Raspberry Pi yang bisa langsung terhubung ke internet menggunakan WIFI. Sistem ini mungkin mengalami masalah transmisi data jika area sistem tidak memiliki akses WIFI. Sementara penelitian [13] menggunakan esp32 yang terhubung ke modem 4G untuk mengakses internet. Permasalahan yang dapat terjadi pada sistem ini adalah kegagalan pengiriman data yang disebabkan oleh terbatasnya jangkauan operator seluler. Jika perangkat harus diletakkan di area yang belum memiliki akses internet, maka koneksi ke jaringan internet dapat menimbulkan masalah serius.

Masalah jangkauan internet yang terbatas dapat diatasi dengan menggunakan topologi *star*. Dengan masing-masing sensor node akan terhubung dengan sebuah *gateway* yang dapat ditempatkan di area yang memiliki akses internet. Penelitian [4][15][16] menggunakan teknologi WSN berdasarkan NRF24L01 untuk komunikasi antar perangkat. Penelitian [15] menguji kapabilitas maksimum dari modul NRF24L01 yang dapat mengkomunikasikan secara halus hingga jarak 1 Km. Hal tersebut memungkinkan penempatan sensor *nodes* pada area yang jauh dan menghubungkan mereka menggunakan *gateway*

yang ditempatkan di area yang memiliki akses internet. Penelitian [4][16] menggunakan sistem *gateway* untuk mengirimkan data dari *sensor nodes* ke server. Sistem WSN berdasarkan NRF24L01 juga memiliki kelemahan. Salah satu kelemahannya adalah komunikasi *half-duplex*, yang membuat perangkat tidak mampu bertukar data secara terus menerus. Dimana komunikasi *half-duplex* membutuhkan perangkat berubah dari model pengiriman atau mode penerimaan. Karena perangkat tidak memiliki sistem pewaktuan yang dapat mengatur mode transmisi pengiriman dan penerimaan, terdapat kemungkinan bahwa *sensor node* dan *gateway* akan berada pada mode yang sama sehingga tidak dapat mengirimkan data secara tepat.

Penelitian [12] dilakukan untuk mengendalikan banjir, dengan teknologi yang dapat membuka atau menutup pintu air berdasarkan ketinggian air. Sistem ini masih mengandalkan logika dasar fuzzy dimana pintu air akan terbuka jika terdeteksi air pada kedalaman tertentu. Sistem ini membutuhkan fitur tambahan agar dapat digunakan untuk mengontrol saluran irigasi. Pembukaan dan penutupan saluran irigasi tidak dilakukan hanya berdasarkan tinggi muka air di pintu irigasi tetapi kebutuhan air untuk masing-masing lahan. Selain itu, sistem [12] hanya dapat membuka atau menutup pintu irigasi, tidak ada opsi untuk mengatur jumlah bukaan pintu irigasi.

Berdasarkan kasus-kasus yang telah dikemukakan di atas, maka usulan rumusan masalah yang dijadikan fokus sistem irigasi yang dibangun adalah:

- Bagaimana melakukan komunikasi secara real time dengan latensi minimal
- Bagaimana menghadirkan sistem WSN berbasis NRF24L01 yang dapat berkomunikasi terus menerus tanpa harus bergantian
- Bagaimana sistem pengendalian pintu irigasi berdasarkan kebutuhan air sawah dan ketinggian air di saluran irigasi
- Cara membuka pintu air secara dinamis.

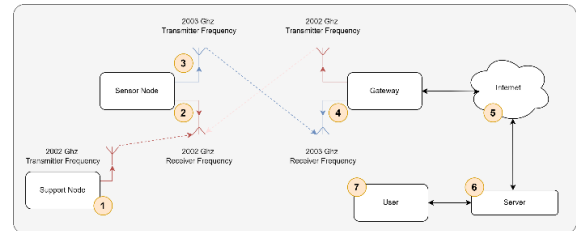
Penelitian ini disusun dalam 4 bab. Bab I membahas tentang latar belakang dan permasalahan sistem irigasi yang ada. Bab II membahas tentang cara kerja sistem dan perangkat yang digunakan. Bab III akan memaparkan hasil implementasi dan pengujian sistem. Terakhir, Bab IV berisi kesimpulan dari penelitian ini.

## II. METODE PENELITIAN

### A. Diagram Sistem

Sistem smart irigasi yang dibuat terdiri dari 3 jenis perangkat keras. Perangkat pertama adalah *support node* yang akan bertanggung jawab untuk merekam kelembaban tanah. Perangkat lain adalah

*node sensor*. *Node sensor* akan merekam data ketinggian air, kecepatan aliran air dan ukuran bukaan pintu air. Selain itu, *node sensor* akan mengontrol seberapa besar bukaan pintu air. Perangkat terakhir adalah *central node* atau *node gateway*, perangkat ini akan menerima koneksi antara *node sensor* dan layanan *cloud*. Sistem irigasi cerdas yang dibangun akan memiliki kemampuan untuk memantau dan mengontrol saluran irigasi dari jarak jauh. Gambar 1 menunjukkan pengiriman data untuk kebutuhan pemantauan irigasi.

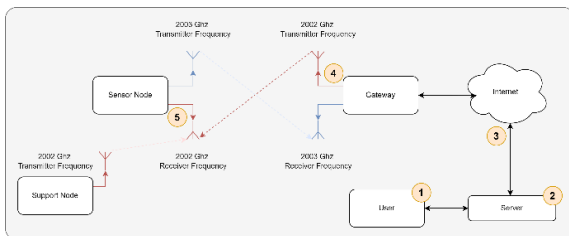


Gambar 1. Aliran data

Alur transmisi data untuk fungsi *monitoring* diawali dengan (1) *support node* akan merekam data kelembaban tanah dan akan mengategorikannya menjadi dua kategori yaitu membutuhkan air atau tidak membutuhkan air. Setelah data terekam maka data akan dienkripsi dan akan dikirimkan ke *sensor node* dengan frekuensi yang telah ditentukan. (2) *Sensor node* akan melakukan pendeskripsian dan validasi data, jika data sudah valid maka data akan dikirim menggunakan komunikasi serial ke mikrokontroler yang bertugas mengirimkan data. (3) Mikrokontroler yang bertugas mentransmisikan data akan mengenkripsi data yang akan dikirim, gelombang frekuensi yang digunakan adalah frekuensi penerima pada node pusat. (4) Data yang telah dikirimkan akan sampai di *gateway* dan dideskripsikan serta divalidasi. (5) Apabila data dinyatakan valid maka data tersebut akan diteruskan ke server dengan menggunakan jaringan internet. Transmisi data dari *gateway* ke server akan menggunakan protokol *WebSocket*. (6) Data yang diterima server akan divalidasi, administrator dapat mengatur interval penyimpanan data ke database, jika interval waktu terpenuhi, data dapat disimpan di database, jika tidak, data akan disiarkan kembali ke halaman pengguna. (7) Pengguna dapat langsung melihat data rekaman sensor melalui website.

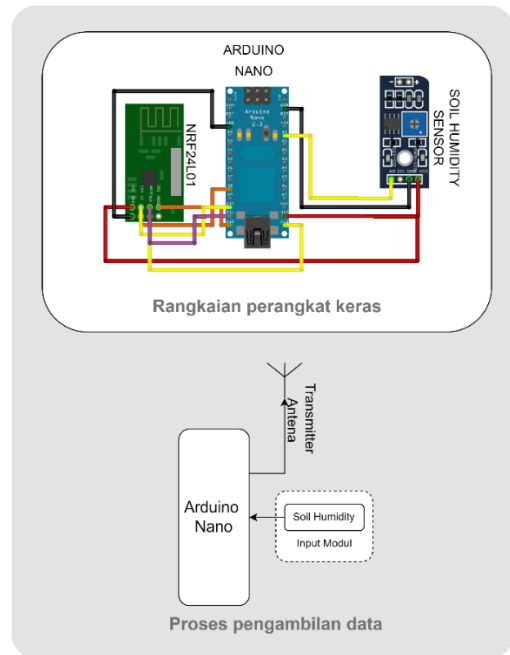
Alur transmisi data untuk fungsi pengaturan situs web dijelaskan pada Gambar 2 (1) Pengiriman data diawali oleh pengguna yang mengunjungi situs web sistem irigasi cerdas, administrator situs web dapat memilih pintu mana yang akan diatur dan ukuran bukaan pintu. (2) Data selanjutnya akan dikirimkan ke server dan (3) diteruskan ke *central node* menggunakan jaringan internet dengan protokol *WebSocket*. (4) *Gateway* akan memvalidasi apakah

data ditujukan ke *gateway* yang tepat, jika data berhasil divalidasi, data akan dienkripsi dan dikirimkan menggunakan gelombang radio ke *sensor node*. (5) *Sensor node* yang menerima data akan mendeskripsikan dan memvalidasi data tersebut. Jika data berhasil divalidasi maka sensor node akan membuka pintu irigasi berdasarkan perintah yang dilakukan oleh administrator sistem.



Gambar 2. Aliran data untuk kontrol

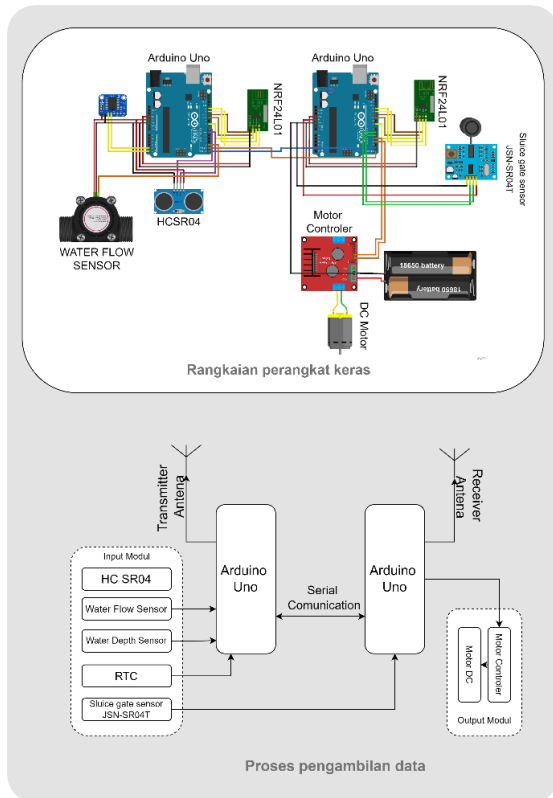
Sistem *smart* irigasi memiliki layanan *cloud*, salah satu layanan yang disediakan adalah website yang dapat diakses melalui link <https://smartrigation.herokuapp.com/>. Situs web ini dibuat dengan bantuan teknologi Nodejs. Nodejs adalah perangkat lunak yang berfungsi sebagai runtime untuk JavaScript. Situs web ini dibangun dengan bantuan kerangka kerja ExpressJS. ExpressJS adalah kerangka kerja yang bekerja di Google V8 Engine, ExpressJS menyederhanakan dan mempercepat pengembangan aplikasi[17]. Layanan lain yang disediakan oleh layanan *cloud* sistem irigasi cerdas adalah API yang digunakan setiap *gateway* untuk berkomunikasi dengan server. API yang digunakan berjalan pada protokol *WebSocket*. Layanan *cloud* menggunakan pustaka Socket.io di NodeJS untuk membangun fungsionalitas yang berjalan di *WebSocket*. Sistem *cloud service* menggunakan PostgreSQL untuk sistem penyimpanan datanya, PostgreSQL terhubung dengan sistem dengan bantuan Prisma ORM (*Object Relation Mapping*). Prisma adalah salah satu ORM modern yang tersedia untuk *runtime* NodeJS.



Gambar 3. Struktur dari *support node*

### B. Pengembangan Perangkat Keras

Sistem irigasi pintar memiliki 3 jenis perangkat keras. Setiap jenis perangkat keras memiliki fungsi dan modul pendukung yang berbeda. Perangkat keras pertama adalah simpul pendukung. Simpul pendukung merupakan perangkat pendukung yang bertugas merekam data kelembaban tanah sawah. Gambar 3 merupakan rangkaian perangkat pada *support node*. Mikrokontroler yang digunakan pada *support node* adalah Arduino Nano. Arduino nano dipilih karena ukurannya yang kecil dan *board* tersebut telah terintegrasi dengan rangkaian FT232RL tambahan berupa *interface* USB yang memudahkan komunikasi dengan komputer [18]. Untuk mengukur kelembaban tanah, alat yang digunakan adalah sensor kelembaban tanah. Sensor ini menggunakan IC LM393 yang memungkinkan sensor menghasilkan data analog atau data digital. Data analog merupakan nilai satuan dari 0 sampai 1023 yang dapat digunakan untuk mengategorikan kondisi basah atau kering. Sensor juga mendukung sinyal digital yang dapat memberikan informasi berupa 1 atau 0, atau sensor dalam kondisi basah atau kering, nilai batas kering dan basah dapat diatur dengan memutar potensiometer pada modul.

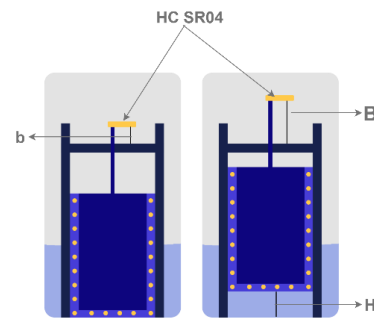


Gambar 4. Struktur dari *sensor node*

Perangkat keras kedua adalah *sensor node*. *Sensor node* seperti terlihat pada Gambar 4 merupakan sensor yang bertugas merekam kedalaman air, kecepatan aliran air dan besar kecilnya bukaan pintu air. Selain bertugas merekam data, sensor node bertugas mengendalikan motor DC yang bertugas membuka dan menutup pintu air. Mikrokontroler yang digunakan untuk sensor node adalah Arduino Uno (ATmega 328P). Perangkat ini dipilih karena bersifat *open source*, didukung bahasa pemrograman C yang cepat, serta memiliki *bootloader* yang mudah dioperasikan. [19]. Untuk mengukur kedalaman air sensor yang digunakan adalah HC SR04, sensor ini mengandalkan gelombang ultrasonik untuk mengukur jarak suatu objek. Sensor tersebut dapat mendeteksi objek dengan rentang jarak 2cm hingga 500cm [20], dengan margin kesalahan pengukuran hingga 2,1% [4]. Pengukuran kecepatan arus dilakukan dengan sensor aliran air dengan tipe YF-S201 ini memiliki sensor *pinwheel* di dalamnya untuk mengukur seberapa banyak air yang melewatinya dan terintegrasi dengan sensor *magnetic hall-effect* yang menghasilkan pulsa elektrik pada setiap putarannya. [21]. Sensor ini bekerja menggunakan prinsip elektromagnetik dimana ketika air mengalir melalui sensor, aliran tersebut akan mempengaruhi sirip turbin sehingga roda berputar. Poros roda turbin dihubungkan

dengan sensor *hall effect* sehingga dihasilkan pulsa setiap putaran dan pulsa ini dapat dipantau melalui mikrokontroler dan digunakan untuk menentukan volume air yang melewatinya. [22].

Pengukuran ukuran bukaan pintu air dilakukan dengan HC SR04. Sensor akan ditempatkan di bagian atas poros *lift* pintu air seperti yang ditunjukkan pada Gambar 5. Untuk menentukan ukuran bukaan, rumus (1).

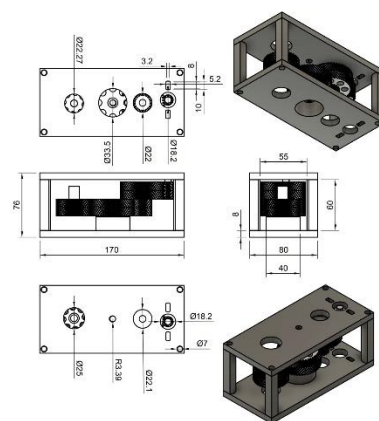


Gambar 5. *Irrigation floodgates*

Pengiriman data pada setiap node menggunakan modul NRF24L01. Modul ini bekerja dengan frekuensi radio 2,4 GHz dengan 125 pilihan band frekuensi, modul ini bekerja pada tegangan 3v, dan mengonsumsi arus sebesar 9 mA pada keluaran daya -6dBm dan 12,3 mA pada mode Rx, modul ini menggunakan metode komunikasi SPI [23].

$$H = B - b \quad \square \square$$

Variabel *b* adalah tinggi poros saat pintu ditutup (tinggi alas), *B* adalah tinggi poros saat pintu dibuka, dan *H* adalah ukuran bukaan pintu irigasi.



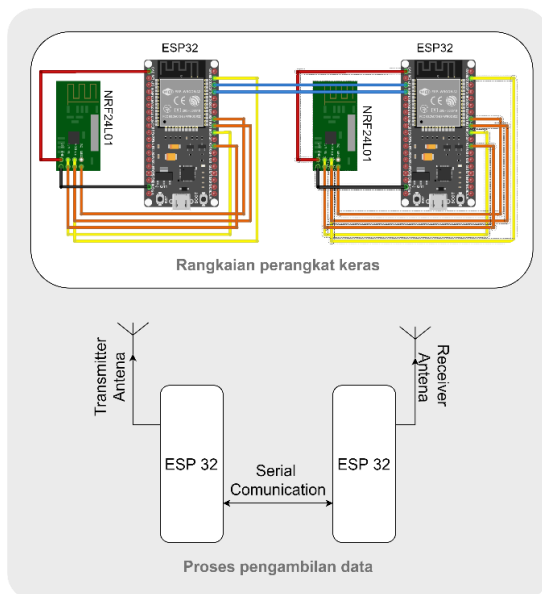
Gambar 6. Rancangan *gearbox*

Node sensor akan memiliki mekanisme untuk membuka atau menutup pintu secara dinamis. Tingkat bukaan pintu dapat diatur secara otomatis menggunakan kecerdasan buatan atau secara manual melalui website. Persamaan (1) menunjukkan



bagaimana pintu dapat dibuka secara dinamis. Sensor akan mengukur ketinggian poros pada saat pintu ditutup, untuk mencapai tingkat keterbukaan yang diinginkan motor harus menggerakkan poros sejauh jarak yang diinginkan ditambah ketinggian alas poros. Dalam pembuatannya, penulis menggunakan mekanisme roda gigi heliks ganda yang dapat dilihat pada Gambar 6.

Perangkat keras ketiga adalah simpul pusat. Node pusat atau gateway seperti pada Gambar 7 merupakan penghubung antara sensor node dan server. *Gateway* dapat diartikan sebagai penghubung komunikasi dalam sistem yang menggunakan protokol yang berbeda, oleh karena itu *gateway* memiliki fungsi yang sama yaitu mengubah protokol [16]. Pada sistem smart irigasi, *gateway* akan menghubungkan komunikasi berbasis frekuensi radio dengan komunikasi berbasis TCP/IP. Node pusat menggunakan modul NRF24L01, untuk berkomunikasi dengan node sensor. Komunikasi ke server berlangsung menggunakan jaringan internet, hal ini dapat terjadi karena *central node* menggunakan ESP32, yaitu mikrokontroler yang memiliki kemampuan untuk mengakses internet melalui WIFI. ESP32 adalah mikrokontroler yang sangat kuat, berdasarkan Xtansa LX6 SoC (*System on Chip*), ESP32 dipilih karena kelebihanannya yang memiliki pin out lebih banyak (36 pin yang dapat digunakan) dan 16 pin analog, memori *flash* besar (total 4Mb, tersedia 512 *byte* untuk digunakan), *base clock* yang lebih tinggi, dan dilengkapi dengan *low power bluetooth 4.2* dan *WIFI 802.11 b/g/n* yang memungkinkan untuk diimplementasikan pada perangkat IoT [24][25].



Gambar 7. Struktur dari *central node*

### C. Pengembangan Perangkat Lunak

Layanan website menyediakan fungsi untuk menampilkan data secara *real time*, dan data sensor yang telah direkam sebelumnya. Website juga berfungsi sebagai media pengaturan sistem irigasi. Sistem irigasi cerdas dapat bekerja secara otomatis menggunakan kecerdasan buatan, sistem juga dapat dikendalikan secara manual dari jarak jauh. Layanan situs web dibangun menggunakan teknologi JavaScript, yang berjalan di NodeJS. Nodejs adalah runtime yang memungkinkan JavaScript bekerja di luar browser, dibangun di atas mesin Chrome v8 dan mengimplementasikan I/O non-pemblokiran berbasis peristiwa, yang memungkinkan sistem ini menjadi sangat ringan dan efisien serta cocok untuk penggunaan waktu nyata [26]. JavaScript adalah bahasa pemrograman tingkat tinggi dan menggunakan skema penulisan dinamis yang berarti pengembang tidak perlu menentukan tipe data tertentu, dan mendukung pemrograman berorientasi objek atau paradigma pemrograman fungsional. [27][28]. JavaScript memiliki beberapa keunggulan dibandingkan bahasa pemrograman lain, penelitian [27] menunjukkan JavaScript sebagai bahasa pemrograman yang sangat cepat dan efisien untuk proyek skala kecil hingga menengah, mengungguli Java dan Python.

Situs web ini dibangun dengan bantuan kerangka kerja ExpressJS. Riset [17] menyatakan bahwa JavaScript menggunakan CPU 2 kali lebih sedikit daripada ASP .NET, JavaScript juga 3 kali lebih efisien dalam penggunaan memori dan ExpressJS dapat memberikan waktu respons yang lebih cepat daripada ASP .NET. Data hasil rekaman sensor akan disimpan menggunakan PostgreSQL. PostgreSQL adalah program sistem manajemen basis data objek-relasional (ORDBMS) yang diperkenalkan sebagai proyek sumber terbuka yang banyak digunakan di bidang kesehatan, astronomi, dan perdagangan, ditulis dalam bahasa C dan menggunakan model multi proses untuk mendukung akses bersamaan yang tinggi [29]. PostgreSQL adalah pilihan terbaik untuk proyek skala kecil hingga menengah. Tes yang dilakukan di [30] menunjukkan bahwa PostgreSQL memiliki kinerja terbaik untuk proses yang melibatkan 100-500 *query* penyisipan dengan hanya diperlukan rata-rata 383,22 md, mengungguli MySQL (470,25 md) dan MongoDB (612,56 md) [30]. Hal ini sangat cocok untuk kebutuhan sistem irigasi dimana proses memasukkan atau merekam data lebih sering dilakukan daripada proses membaca data.

Layanan website akan mengategorikan tingkat air menjadi 3 tingkatan [4] yaitu rendah, menengah dan tinggi. Sedangkan kecepatan air akan dikategorikan menjadi 3 tingkatan juga [31] yaitu rendah, menengah dan cepat. Sedangkan disisi lain

kebutuhan air akan diwakili oleh sensor dimana 1 berarti tanah kering dan membutuhkan air sedangkan 0 berarti tanah basah dan tidak membutuhkan air. Hal ini dilakukan untuk memudahkan user memahami status dan kondisi di lapangan. Tabel 1 menunjukkan aturan fuzzy untuk mengategorikan data rekaman sensor.

TABEL I. FUZZY RULES UNTUK KATEGORISASI DATA

Data Type	Condition	Status
Water Level	IF water depth < 30% maximum limit THEN	Low
	IF water depth >= 30% AND water depth < 80% maximum limit THEN	Medium
	IF water depth >80% maximum limit THEN	High
Water Flow Speed	IF current <25cm/s THEN	Slow
	IF current > 25cm/s AND <50cm/s THEN	Medium
	IF current >50cm/s THEN	Fast

Selain menyediakan halaman antarmuka pengguna, layanan web juga menyediakan API Platform untuk konektivitas antar perangkat. Dengan menyediakan platform API, node pusat (*gateway*) dapat meneruskan data yang diterima dari sensor node ke server. Platform API berjalan di atas protokol *WebSocket*. Kurangnya kemampuan komputasi pada perangkat mikrokontroler menyebabkan protokol http tidak dapat bekerja dengan baik. Berdasarkan penelitian [1] [4] sebuah sistem yang menggunakan protokol http memerlukan 1-4 detik untuk menyelesaikan pengiriman. Penelitian [7] menunjukkan penggunaan *WebSocket* sebagai protokol komunikasi antara mikrokontroler dan server dapat bekerja dengan baik dan hanya memerlukan kurang dari 300ms untuk setiap transaksi.

TABEL II. DATASET DECISION TREE

Water Level	Water Needs	Water Current Speed	Irrigation Door Opening
Low	Low	Slow	Low
Medium	Low	Slow	Low
High	Low	Slow	Low
Low	High	Slow	Low
Medium	High	Slow	Medium
High	High	Slow	Medium
Low	Low	Medium	Low
Medium	Low	Medium	Low
High	Low	Medium	Low
Low	High	Medium	Low
Medium	High	Medium	Medium
High	High	Medium	High
Low	Low	Fast	Low
Medium	Low	Fast	Medium
High	Low	Fast	High
Low	High	Fast	Medium
Medium	High	Fast	Medium

Water Level	Water Needs	Water Current Speed	Irrigation Door Opening
High	High	Fast	High

Kecerdasan buatan digunakan untuk mengetahui proses membuka dan menutup pintu air secara otomatis. *Decision tree* merupakan metode pengambilan keputusan berdasarkan pohon keputusan, pohon keputusan tersebut menggunakan skema algoritma pengambilan keputusan berantai yang disusun menjadi suatu hierarki keputusan. Untuk melakukan proses pembukaan dan penutupan pintu air, ada 3 parameter yang akan digunakan untuk mengambil keputusan akhir. Parameter yang digunakan adalah ketinggian air, jumlah air yang dibutuhkan dan kecepatan arus yang diukur menggunakan water flow sensor. Tabel 2 adalah model kecerdasan buatan [32].

Tabel 2 merupakan *dataset* yang akan digunakan untuk model kecerdasan buatan. Kecerdasan buatan yang akan digunakan dapat memanfaatkan algoritma Gini, Entropy, Log Loss untuk memprediksi tingkat bukaan pintu air. Selain algoritma tersebut, ada faktor lain yang dapat mempengaruhi akurasi prediksi, faktor tersebut adalah metode *splitter*, metode yang dapat digunakan adalah metode terbaik atau acak. Prediksi yang dihasilkan berupa *range* perintah untuk membuka beberapa pintu air (rendah), membuka beberapa pintu air (sedang), dan membuka sebagian besar pintu air (tinggi). Fitur dari model ini adalah ketinggian air, kebutuhan air dan kecepatan arus air. Ketiga fitur ini akan mempengaruhi tingkat keterbukaan pintu air. Misalnya, jika ketinggian air tinggi, dan sawah membutuhkan banyak air dan arus air menunjukkan kecepatan sedang, maka pintu air dibuka lebar. Program kecerdasan buatan tidak akan dapat mengenali kategori fitur dalam bentuk *string*, sehingga diperlukan *encoder* label untuk mengubah data ini. *Encoder* label akan mengubah data menjadi bilangan bulat seperti yang ditunjukkan pada Tabel 3.

TABEL III. POST-LABELLED DATASET

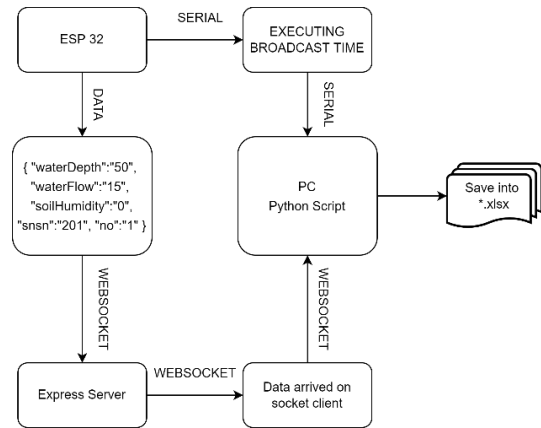
Water Level (Features 1)	Water Needs (Feature s 2)	Water Current Speed (Features 3)	Irrigation Door Opening (level)
0	0	0	0
1	0	0	0
2	0	0	0
0	2	0	0
1	2	0	1
2	2	0	1
0	0	1	0
1	0	1	0
2	0	1	0
0	2	1	0

Water Level (Features 1)	Water Needs (Features 2)	Water Current Speed (Features 3)	Irrigation Door Opening (level)
1	2	1	1
2	2	1	2
0	0	2	0
1	0	2	1
2	0	2	2
0	2	2	1
1	2	2	1
2	2	2	2

### III. HASIL DAN PEMBAHASAN

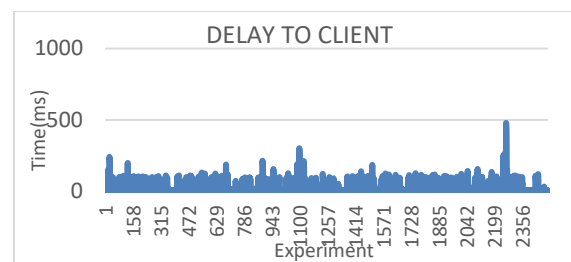
Dalam teknologi *Internet of Things* (IoT), terdapat 4 komponen utama yang mendukung sistem tersebut. Komponen tersebut adalah sensor, aplikasi, infrastruktur jaringan, dan layanan backend. Tes umum yang digunakan untuk IoT adalah kegunaan, keandalan, integritas data, keamanan, dan kinerja. Pada penelitian ini pengujian akan difokuskan pada aplikasi berupa penggunaan website, infrastruktur jaringan dan layanan backend. Pengujian yang akan dilakukan berupa pengujian *usability*, *reliability*, *connectivity* dan *performance*. Semua pengujian dilakukan menggunakan server lokal untuk menghilangkan faktor kualitas transmisi jaringan.

Pengujian *usability* meliputi pengujian mekanisme pintu air, pengujian ini dilakukan untuk mengetahui kemampuan alat dalam mengangkat pintu air. Pengujian *reliability* dilakukan untuk mengetahui kemampuan suatu sistem atau komponen untuk menjalankan fungsi tertentu dalam waktu tertentu. Pengujian *reliability* meliputi pengujian *firmware*. Pengujian *performance* bertujuan untuk mengetahui kecepatan dan ketepatan pemrosesan, jumlah pengguna yang dapat terhubung dalam satu waktu dan waktu respons. Pengujian *performance* akan mencakup pengujian AI. Pengujian *connectivity* dilakukan untuk memastikan transmisi data antar perangkat berjalan dengan baik dan dengan latensi rendah. Pengujian *connectivity* akan mencakup pengujian komunikasi radio.



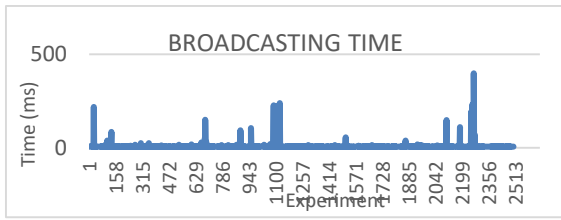
Gambar 8. Skenario pengujian *WebSocket*

Pengujian pertama adalah pengujian *Firmware* dan kemampuan perangkat keras untuk menguji keandalan perangkat. Pengujian seperti pada Gambar 8 dilakukan dengan melakukan *stress test* pada perangkat ESP 32. ESP 32 akan terhubung ke jaringan dan akan mengirimkan beberapa data menggunakan teknologi *WebSocket*. Lamanya waktu pengiriman atau eksekusi (*broadcasting*) data dari ESP 32 akan dikirim menggunakan komunikasi serial ke komputer, terdapat *script* yang dibuat dengan Python yang akan merekam data tersebut dan menyimpannya dalam *file excel*. Skrip python juga akan menjadi klien socket untuk mengukur waktu tunda agar data tiba/muncul di halaman klien. Pengujian akan dilakukan dalam 4 *batch*; setiap *batch* akan mengirimkan data yang akan ditandai sebagai rekaman sensor. Setiap data berisi ketinggian air, kecepatan arus, kelembaban tanah, sensor ID dan nomor untuk mengidentifikasi urutan data yang dimasukkan. Tes *batch* 1 akan berisi 2500 data yang akan dihasilkan secara acak oleh ESP 32. *Batch* 2 akan terdiri dari 5000 data, *batch* 3 akan terdiri dari 7500 data dan *batch* terakhir akan terdiri dari 10000 data.

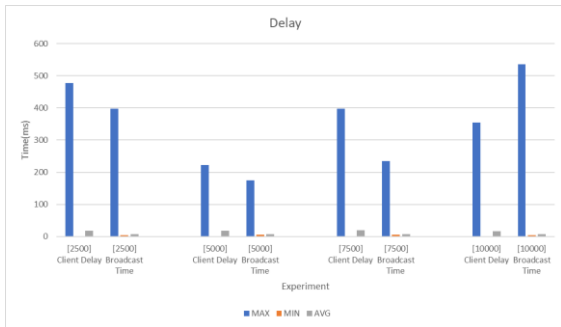


Gambar 9. *Delay* untuk klien





Gambar 10. Broadcasting time ESP32

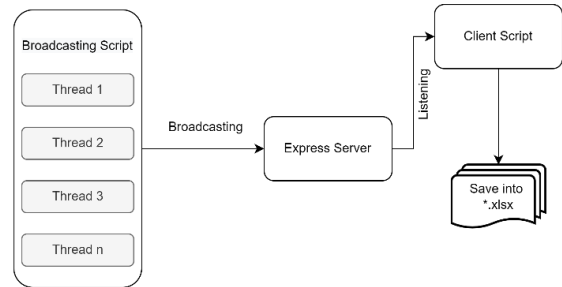


Gambar 11. Semua penundaan batch broadcasting

Gambar 9 menunjukkan fluktuasi waktu saat data diterima oleh klien. Rata-rata waktu yang dibutuhkan setiap data untuk diterima oleh klien adalah 18,44 ms, dengan waktu tercepat 1 ms dan waktu terlama mencapai 478 ms. Gambar 10 menunjukkan lama waktu *Broadcasting* 2500 data yang dibutuhkan oleh ESP 32. ESP 32 membutuhkan waktu rata-rata 7,84ms, dengan waktu tercepat untuk mengeksekusi tugas *Broadcasting* adalah 4ms dan waktu terlama yang dibutuhkan adalah 397ms. Selama transmisi 2500 data terjadi peningkatan suhu pada *board* ESP 32. Namun demikian, ESP 32 tetap dapat mengirimkan data secara konsisten meskipun terjadi anomali waktu pengiriman yang lebih lama yang meningkat pada akhir *batch* 1. Data yang dikirimkan oleh ESP akan disiarkan ulang oleh server ke klien yang terhubung. Gambar 11 menunjukkan hasil pengujian secara keseluruhan, konsistensi waktu tunda sehingga data dapat diterima oleh klien sebagian besar berada di bawah ambang batas 200 ms.

Untuk *batch* 2 data yang akan dikirim adalah 5000 data. Dengan waktu siaran rata-rata 6,8 ms. Untuk *batch* 3 rata-rata waktu tayang adalah 6,96 ms, sedangkan untuk *batch* terakhir rata-rata waktu yang dibutuhkan adalah 7,35 ms. Dari data tersebut, semakin banyak data yang dikirimkan, maka semakin tinggi nilai rata-rata yang dibutuhkan untuk menyiarkan data. Waktu minimum untuk penyiaran pada gelombang 2-4 adalah 5ms. Sama dengan hasil

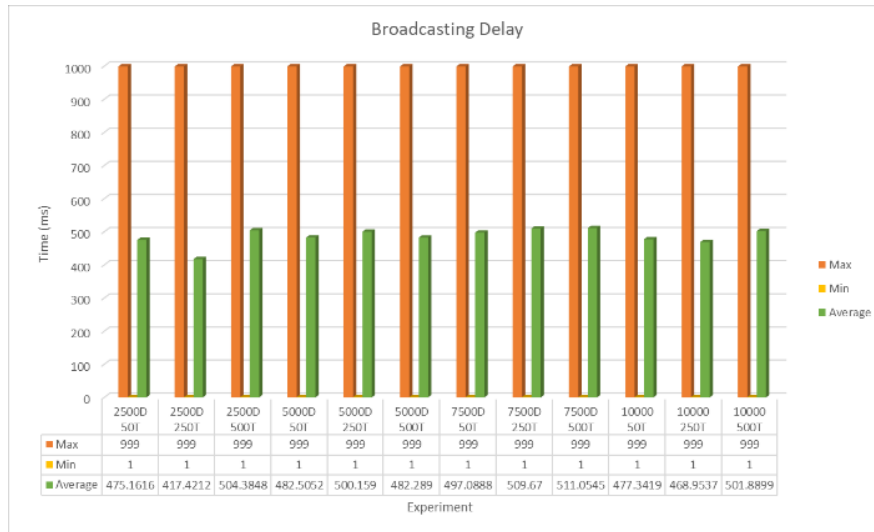
pengujian *batch* 1, waktu tayang terlama terus bertambah seiring dengan jumlah data yang dikirim. Sedangkan waktu tunda rata-rata data muncul di halaman klien adalah *batch* 2 18,67ms, *batch* 3 19,63ms, *batch* 4 16,27. Banyaknya data tidak terlalu mempengaruhi waktu pengiriman data agar dapat diterima oleh *client*. *Batch* 2-4 memiliki waktu tunda di bawah 200ms.



Gambar 12. Skema pengujian server WebSocket

Tes kedua adalah tes *WebSocket*. *WebSocket* memiliki kemampuan mengirim data lebih cepat dari http. Selain itu, *WebSocket* dapat langsung mengirimkan data ke klien yang terhubung secara real time. Tes pertama membuktikan bahwa teknologi *WebSocket* dapat mengirim data dari ESP 32 ke *client* dengan waktu rata-rata 18ms. Untuk menguji performa maksimal dan keandalan layanan backend, akan dilakukan pengujian dengan memanfaatkan teknologi *threading*. Gambar 12 menunjukkan skema pengujian keandalan *server*. *Threading* akan menstimulasikan jumlah perangkat yang terhubung pada satu waktu. Skrip pengujian dibuat menggunakan python dan hasil pengujian akan disimpan dalam *file excel*. Pengujian ini dilakukan untuk mengetahui berapa lama waktu yang dibutuhkan untuk mengirimkan data dari *broadcaster* ke *client*. Pengujian dilakukan dengan skema 2500, 5000, 7500 dan 10000 data. Setiap *batch* data akan diuji sebanyak 3 kali menggunakan 50, 250 dan 500 *thread*.

Gambar 13 menunjukkan hasil pengujian server *WebSocket*. Dari hasil pengujian, waktu terlama suatu data dapat diterima oleh klien adalah 999ms dan waktu tercepat adalah 1ms. Hasilnya sama di seluruh pengujian dari kisaran 2500 hingga 10.000 data. Jumlah utas juga tidak mempengaruhi waktu maksimum atau minimum. Waktu rata-rata setiap *batch* adalah 417ms hingga 511ms. Banyaknya perangkat yang terhubung ke server tidak mengurangi kinerja layanan *backend*.

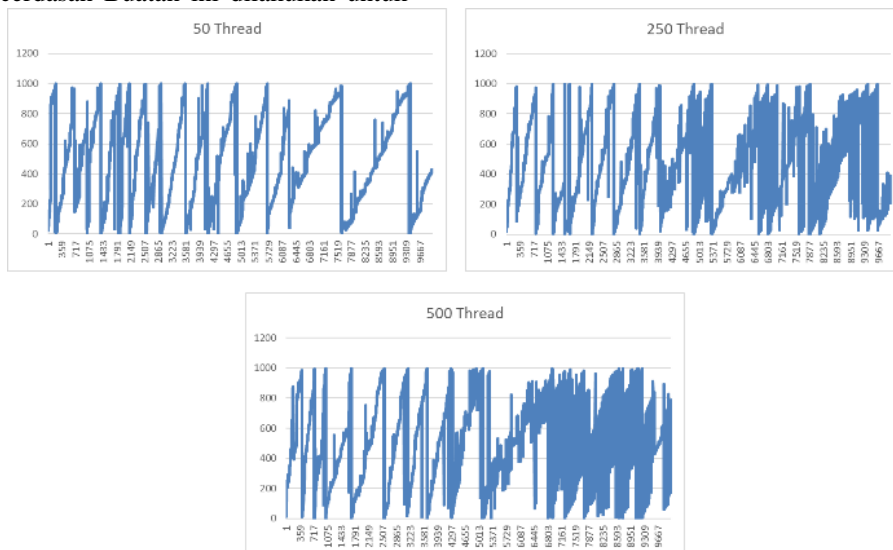


Gambar 13. Kinerja Server WebSocket

Gambar 14 menunjukkan grafik lama waktu pengiriman data. Pengujian menggunakan 10.000 data dan menggunakan 50, 250, dan 250 utas. Grafik tersebut memiliki jumlah pola yang sama. Waktu transmisi akan bertambah saat data pertama dikirim. Setelah mencapai titik tertinggi (999ms), waktu transmisi akan berkurang lagi. Setelah siklus selesai, waktu transmisi akan bertambah lagi. Dan waktu transmisi tidak akan melebihi 999ms.

Tes ketiga adalah tes kecerdasan buatan. Pengujian Kecerdasan Buatan ini dilakukan untuk

mengetahui kecepatan dan keakuratan program, untuk mengeksekusi setiap permintaan. Pengujian pertama dilakukan untuk mengetahui tingkat akurasi prediksi. Pada pengujian pertama ada dua skema yang akan dilakukan. Skema pertama yang akan dilakukan adalah pengujian akurasi dimana *dataset* yang dilatih tidak sama dengan *dataset* yang akan diuji. Skema kedua adalah akurasi pelatihan dimana *dataset* yang akan dilatih dan data yang akan diuji adalah data yang sama.



Gambar 14. Fluktuasi Server Websocket

Pengujian akurasi akan membagi *dataset* pada Tabel 3 menjadi dua bagian. Bagian pertama adalah data yang akan digunakan sebagai model pelatihan, data ini berisi 14 data dari total 18 data yang tersedia dengan komposisi 7 data level 0, 5 data level 1 dan 2 data level 2. Data yang digunakan

sebagai data pengujian terdiri dari 2 data level 0 dan masing-masing 1 data untuk level 1 dan 2. Ada beberapa faktor yang akan mempengaruhi tingkat akurasi program diantaranya algoritma dan metode *splitter*. Pengujian akan dilakukan dengan menggabungkan faktor-faktor tersebut. Setiap

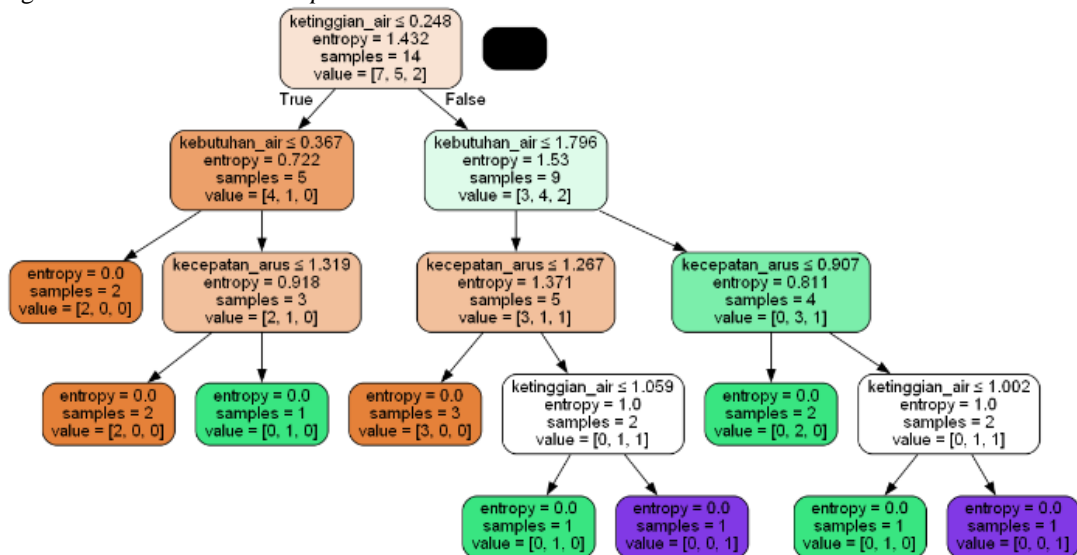
kombinasi algoritma dan metode *splitter* akan diuji sebanyak 10 kali dan akan dihitung rata-ratanya.

TABEL IV. HASIL UJI AKURASI

Algorithm	Splitter	Avg accuracy (%)	Avg training time (ms)
Entropy	Random	55	0,99
	Best	50	1,37
Gini	Random	52	0,99
	Best	50	0,94
Log Loss	Random	50	0,95
	Best	50	1,13

Tabel 4 menunjukkan bahwa algoritma *Entropy* dan metode *random splitter* menghasilkan akurasi tertinggi dengan rata-rata akurasi 10 tes mencapai 55% dengan rata-rata waktu yang dibutuhkan untuk melakukan pelatihan selama 0.99ms. Algoritma *Gini* dan metode *splitter* terbaik

adalah algoritma tercepat dengan catatan rata-rata waktu pelatihan 0,94ms. Metode *splitter* terbaik yang dikombinasikan dengan algoritma adapun menghasilkan akurasi 50% yang tetap konstan di setiap pengujian. Algoritme *Log Loss* terus menunjukkan tingkat akurasi 50% di setiap pengujian bila digabungkan dengan metode *splitter* terbaik atau acak. Algoritma *Gini*, dikombinasikan dengan metode *random splitter*, menghasilkan akurasi yang berfluktuasi antara 25% dan 75% selama pengujian. Algoritma *entropy* yang dikombinasikan dengan metode *random splitter* menunjukkan hal yang serupa, terdapat fluktuasi akurasi selama pengujian, akurasi berkisar 25% dan mencapai nilai tertinggi 100%, pohon keputusan pohon yang dihasilkan oleh algoritma ini dapat dilihat melalui Gambar 15.



Gambar 15. Entropi Algoritma Pohon Keputusan dan Pemisah Acak

Pengujian ini dilakukan dengan cara *up sampling* dari *dataset* yang tersedia. *Dataset* yang akan diuji berisi 4800 sampel data dengan komposisi 2394 level 0, 1596 level 1 dan 809 level 2. Setiap sampel data memiliki status level air (dengan skala 0,1,2), level kebutuhan air (dengan pilihan data 0 atau 2) serta tingkat kecepatan air (dengan skala 0,1,2). Hasil pengujian berupa nilai akurasi dengan rentang 0 hingga 100%. Hasil pengujian menunjukkan bahwa waktu pelatihan yang dibutuhkan adalah 0,6 ms. *Dataset* uji berhasil dieksekusi dalam waktu 1,9ms dengan akurasi prediksi mencapai nilai 100% atau program dapat memprediksi kondisi dengan sangat akurat.

Pengujian kedua adalah pengujian kinerja web server kecerdasan buatan. Web server digunakan agar *webservice* yang dibuat

menggunakan javascript dapat mengakses hasil prediksi yang dibuat dengan python. Semakin banyak utas yang digunakan akan menyimulasikan jumlah pengguna yang mengirimkan permintaan ke server. Pengujian dilakukan dengan data 1000, 3000 dan 5000 menggunakan skema utas 50, 250 dan 500.

Hasil pengujian kinerja server AI dapat dilihat pada Tabel 5, penambahan jumlah data yang diuji tidak berpengaruh signifikan terhadap waktu *respons* minimum. Waktu *respons* minimum untuk setiap penambahan data relatif meningkat dengan peningkatan yang tidak drastis. Semakin banyak utas yang digunakan, semakin tinggi waktu *respons* minimum. Seperti dalam kumpulan 3000 data di mana waktu *respons* minimum pada 50 utas adalah 470 md, itu meningkat menjadi 483 md pada 500 utas. Dalam kumpulan 5000 data, waktu *respons*

maksimum telah berkurang untuk 500 utas dari 15751ms menjadi 2525ms. Penurunan ini disebabkan karena server tidak dapat merespons semua permintaan yang masuk. Jumlah *thread* yang banyak akan mengurangi keberhasilan proses *request* yang disebabkan ketidakmampuan server untuk merespons semua *request* karena beban program tidak terdistribusi ke server lain.

TABEL V. WAKTU RESPONS SERVER AI

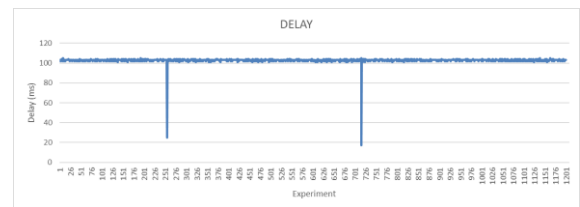
Data Size	Thread Size	Min	Max	Avg
1000	50	473	1495	619
	250	490	1644	645
	500	477	1877	672
3000	50	470	1541	544
	250	474	1508	551
	500	483	3882	1002
5000	50	469	2795	671
	250	526	15715	1113
	500	504	7525	756

Salah satu permasalahan yang terjadi pada penelitian sebelumnya adalah tingginya *delay time* yang terjadi karena perangkat komunikasi harus bergantian mengirim atau menerima data. Pada penelitian ini diusulkan sistem komunikasi *full duplex* menggunakan modul NRF24L01. Untuk menguji kemampuan komunikasi ini akan dilakukan dua skema pengujian. Pengujian pertama akan dilakukan dengan mengirimkan 1200 data dari node sensor ke node pusat. Selisih waktu pengiriman dan penerimaan paket data akan ditetapkan sebagai nilai *delay* dalam komunikasi. Skema kedua yang akan digunakan untuk mengirimkan data secara bersamaan dari node sensor ke node pusat dan sebaliknya. Perbedaan waktu saat data diterima di *sensor node* atau *central node* akan merepresentasikan kecepatan pengiriman.

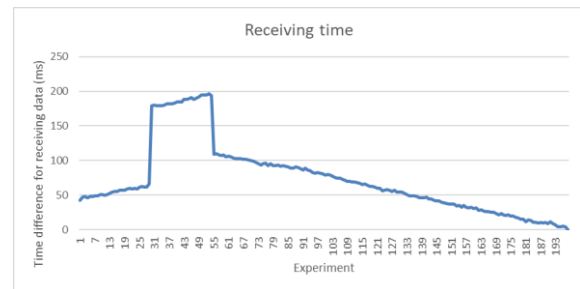
Pengujian skema pertama dilakukan dengan menempatkan dua modul radio pada area tertutup pada jarak 5m. Gambar 16 merepresentasikan *delay* pada setiap pengiriman data. Paket dapat dikirim dalam kerangka waktu rata-rata 103ms. Penelitian mencatat waktu transmisi tercepat 17ms dan waktu terakhir yang dibutuhkan untuk mengirimkan data adalah 105ms. Sistem komunikasi yang dikembangkan memiliki konsistensi yang tinggi dalam kecepatan waktu pengiriman data.

Gambar 17 menunjukkan perbedaan waktu saat data diterima di sensor node atau *central node*. Pengujian membuktikan bahwa data dapat diterima dalam waktu yang bersamaan, hal ini ditunjukkan dengan data yang berhasil diterima pada sensor node dan *central node* dengan selisih waktu 0ms.

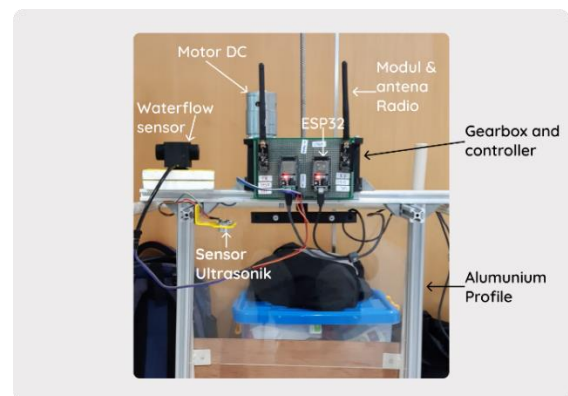
Perbedaan waktu rata-rata adalah 73ms, dan jarak terpanjang adalah 196ms. *Lag* atau selisih yang cukup tinggi pada awal pengujian terjadi karena adanya perbedaan waktu *boot* pada masing-masing ESP 32. Terdapat mikrokontroler yang mengirimkan data terlebih dahulu. Selang beberapa waktu, perbedaan waktu penerimaan data mengalami tren penurunan hingga *delay* tersebut benar-benar hilang. Komunikasi *full-duplex* memungkinkan pengiriman data yang direkam oleh sensor serta instruksi untuk membuka/menutup pintu air untuk dikirim dan diterima pada saat yang bersamaan. Hal ini terjadi karena pada setiap node terdapat modul radio yang bertugas untuk terus mengirimkan data dan modul lainnya yang bertugas untuk terus menerima data. Oleh karena itu, perbedaan waktu penerimaan data dapat dikecilkan.



Gambar 16. Radio Delay



Gambar 17. Full Duplex Communication Delay



Gambar 18. Implementasi Prototipe Sistem Irigasi Cerdas

Gambar 18 menunjukkan implementasi prototipe mekanisme pintu air. *Gearbox* terbuat dari bahan plastik PLA yang dicetak menggunakan printer 3D. Rangka dan rangka pintu air dibuat

menggunakan profil aluminium. Sedangkan daun pintu di replikasi menggunakan akrilik.

Dari hasil pengujian motor DC dapat mengangkat beban akrilik. Tingkat keterbukaan dapat diatur secara dinamis, tidak terbatas pada tingkat terbuka atau tertutup. Mekanisme *gearbox* yang terbuka dan tertutup dari plastik menghasilkan kebisingan yang cukup tinggi. Penggunaan pelumas pada *gearbox* dapat mengurangi tingkat kebisingan dan keausan akibat gesekan, meskipun keausan *gear* tidak dapat dihindari dengan penggunaan yang lama.

#### IV. KESIMPULAN

##### A. Kesimpulan

Berdasarkan hasil pengujian dapat disimpulkan bahwa sistem irigasi cerdas telah berhasil dikembangkan. Hasil tes menunjukkan bahwa:

- Komunikasi antara sensor node dan klien secara *real time* dapat dibuat dengan teknologi *WebSocket*, yang dapat mengirimkan data dan muncul di halaman klien dengan *record* terbaik mencapai 1ms dan waktu rata-rata yang dibutuhkan adalah 417ms. Hasil ini dicapai dengan 500 klien yang terhubung ke jaringan secara bersamaan.
- Sistem komunikasi *full duplex*, memungkinkan *sensor node* dan *central node* untuk terus berkomunikasi tanpa perlu bergantian saat ingin mengirim atau menerima data. Selisih waktu saat data diterima di sensor node atau di *central node* bisa mencapai 0ms, dengan kata lain sistem dapat mengirim dan menerima data secara bersamaan.
- Pengaturan sistem irigasi berdasarkan kebutuhan air untuk tanah dan ketinggian air dapat dibuat dengan teknologi kecerdasan berbasis pohon keputusan dengan *algoritma entropy* dan metode *random splitter*, algoritma tersebut menghasilkan tingkat akurasi 100% rata-rata 55% dari 10 kali lipat tes. Layanan web kecerdasan buatan dapat memberikan waktu respons dengan waktu pemrosesan rata-rata 544ms.

Tingkat kedinamisan bukaan pintu air dapat dicapai dengan menggunakan roda gigi heliks ganda, yang memungkinkan daun pintu air diangkat sesuai kebutuhan.

##### B. Saran

Berdasarkan penelitian yang dilakukan, sistem masih dapat dikembangkan dengan mengganti metode klasifikasi yang lebih baik dibandingkan *decision tree*. Selain itu peningkatan *dataset* untuk pelatihan juga bisa menjadi salah satu

*factor* untuk meningkatkan keakuratan metode klasifikasi.

#### REFERENSI

- [1] L. Kamelia, M. A. Ramdhani, A. Faruqi, and V. Rifadiapriyana, "Implementation of Automation System for Humidity Monitoring and Irrigation System," in *IOP Conference Series: Materials Science and Engineering*, Jan. 2018, vol. 288, no. 1. doi: 10.1088/1757-899X/288/1/012092.
- [2] P. E. Agustyan and A. A. Sabilla, "Rengganis Jurnal Pengabdian Masyarakat Pengelolaan saluran irigasi guna meningkatkan produktivitas pertanian di Desa Jubel Kidul," *Rengganis Jurnal Pengabdian Masyarakat*, vol. 1, no. 2, p. 113, 2021, [Online]. Available: <https://mathjournal.unram.ac.id/index.php/Rengganis/index>
- [3] D. Rusmawan and A. K. dan Muzammil Balai Pengkajian Teknologi Pertanian Kepulauan Bangka Belitung Jalan Mentok, "PENGARUH KETERSEDIAAN AIR TERHADAP PRODUKSI PADI SAWAH," in *Seminar Nasional Hari Air Sedunia*, 2018, pp. 208–214.
- [4] I. Hermawan, D. A. Fachrudin, A. Setiawan, and N. T. S. Sulthanah, "Rancang Bangun Sistem Irigasi Cerdas Menggunakan Metode Fuzzy Rule-Based Untuk Otomatisasi Pintu Air dan Pendeteksian Endapan," *Jurnal Komputer Terapan*, vol. 8, no. 1, pp. 1–11, 2022, [Online]. Available: <https://jurnal.pcr.ac.id/index.php/jkt/>
- [5] Nurhyati, C. R. al Usrah, and Alwi, "Konflik Air Irigasi Antar Petani Sawah di Gampong Tanjong Keumala dan Gampong Babah Buloh Kecamatan Sawang Kabupaten Aceh Utara," *Jurnal Sosiologi Dialektika Sosial*, vol. 1, no. 2, pp. 97–110, 2021.
- [6] M. Jamalaay, J. J. Rikumahu, and D. R. Pattiapon, "AUTOMATISASI PENERAK PINTU AIR PADA BENDUNGAN PERSAWAHAN DI KABUPATEN BURU," *Jurnal Simetrik*, vol. 8, no. 1, pp. 68–76, 2018.
- [7] B. Soewito, Christian, F. Gunawan, Diana, and I. G. P. Kusuma, "Websocket to Support Real Time Smart Home Applications," *International Conference on Computer Science Intelligence*, pp. 560–566, 2019.
- [8] T. Indriyani and M. Ruswiansari, "Kontrol Jarak Jauh Sistem Irigasi Sawah Berbasis Internet Of Things (IoT)," *INTEGER: Journal of Information Technology*, vol. 2, no. 2, pp. 41–48, 2017.
- [9] R. S. Krishnan *et al.*, "Fuzzy Logic based Smart Irrigation System using Internet of Things," *J Clean Prod*, vol. 252, Apr. 2020, doi: 10.1016/j.jclepro.2019.119902.
- [10] S. C. Mungale, M. Sankar, D. Khot, R. Parvathi, and D. N. Mudgal, "An Efficient Smart Irrigation System for Solar System by using PIC and GSM," in *Proceedings of the 5th International Conference on Inventive Computation Technologies, ICICT 2020*, Feb. 2020, pp. 973–976. doi: 10.1109/ICICT48043.2020.9112431.
- [11] D. Prayama, A. Yolanda, and A. W. Pratama, "Rancang Bangun Alat Pengontrol Penyiram Tanaman Otomatis Menggunakan Sensor Kelembaban Tanah Di Area Pertanian," *Jurnal Rekayasa Sistem dan Teknologi Informasi*, vol. 2, no. 3, pp. 807–812, 2018.
- [12] S. S. H. Hajjaj, M. T. H. Sultan, M. H. Mokhtar, and S. H. Lee, "Utilizing the internet of things (IoT) to develop a remotely monitored autonomous floodgate for water management and control," *Water (Switzerland)*, vol. 12, no. 2, Feb. 2020, doi: 10.3390/w12020502.
- [13] M. Farooq, A. Hussain, S. Hashim, L. Yang, and M. Ali, "Automated Irrigation System based on irrigation gates



- using fuzzy logic,” in *2020 International Conference on Internet of Things and Intelligent Applications, ITIA 2020*, Nov. 2020. doi: 10.1109/ITIA50152.2020.9312344.
- [14] S. Sunardi, A. Afif, and F. Noviyanto, “Real Time Monitoring and Irrigation Control Using the WebSocket Protocol,” in *International Council for Scientific and Technical Information*, Apr. 2019. doi: 10.4108/eai.19-10-2018.2282548.
- [15] I. Gusti Made Ngurah Desnanjaya and M. Dwi Alfian, “PENGIRIMAN DATA NRF24L01+ DENGAN KONDISI LINE OF SIGHT DAN NON LINE OF SIGHT,” *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, vol. 3, no. 2, pp. 128–139, 2020, [Online]. Available: <https://s.id/jurnalresistor>
- [16] T. Afif, A. Bhawiyuga, and R. A. Siregar, “Implementasi Perangkat Gateway Untuk Pengiriman Data Sensor Dari Lapangan Ke Pusat Data Pada Jaringan Wireless Sensor Network Berbasis Perangkat nRF24L01,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 4, pp. 3695–3701, 2019, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [17] Oliver Karlsson, “A Performance comparison Between ASP.NET Core and Express.js for creating Web APIs,” *MAIN FIELD: Computer Science*, AUTHOR: Oliver Karlsson SUPERVISOR: Peter Larsson-Green JONKÖPING: July 2021, JONKÖPING UNIVERSITY, JONKÖPING, 2021.
- [18] E. F. Moreno and J. P. Mendoza, “Synergy LEGO Mindstorms – Arduino: Taking advantage of both platforms,” *2018 XIII Technologies Applied to Electronics Teaching Conference (TAAE)*, 2018.
- [19] G. Made, N. Desnanjaya, I. B. Ary, and I. Iswara, “TRAINER ATMEGA32 SEBAGAI MEDIA PELATIHAN MIKROKONTROLER DAN ARDUINO,” Online, 2018. [Online]. Available: <http://jurnal.stiki-indonesia.ac.id/index.php/jurnalresistor>
- [20] F. Puspasari, I. Fahrurrozi, T. P. Satya, G. Setyawan, M. R. al Fauzan, and E. M. D. Admoko, “Sensor Ultrasonik HCSR04 Berbasis Arduino Due Untuk Sistem Monitoring Ketinggian,” *Jurnal Fisika dan Aplikasinya*, vol. 15, no. 2, p. 36, Jun. 2019, doi: 10.12962/j24604682.v15i2.4393.
- [21] A. Ray and S. Goswami, “IoT and Cloud Computing based Smart Water Metering System,” in *2020 International Conference on Power Electronics and IoT Applications in Renewable Energy and its Control, PARC 2020*, Feb. 2020, pp. 308–313. doi: 10.1109/PARC49193.2020.236616.
- [22] M. Tahir, “Measuring water Flow Rate and Volume using Arduino and Flow Sensor By,” Pakistan, 2021.
- [23] A. Septiano and T. Ghozali, “NRF 24L01 SEBAGAI PEMANCAR/PENERIMA UNTUK WIRELESS SENSOR NETWORK,” *Jurnal TEKNO(Civil Engineering, Elektrical Engineering and Industrial Engineering)*, vol. 17, no. 1, pp. 26–34, 2020.
- [24] I. W. Suriana, I. G. A. Setiawan, and I. M. S. Graha, “Rancang Bangun Sistem Pengaman Kotak Dana Punia berbasis Mikrokontroler NodeMCU ESP32 dan Aplikasi Telegram,” *Jurnal Ilmiah Telsinas*, vol. 4, no. 2, 2021.
- [25] M. Babiuch, P. Folytnek, and P. Smutny, “Using the ESP32 Microcontroller for Data Processing,” in *2019 20th International Carpathian Control Conference (ICCC)*, May 2019, pp. 1–6. doi: 10.1109/CarpathianCC.2019.8765944.
- [26] H. Brar, T. Kaur, and Y. Rajoria, “The Better Comparison between PHP, Python-web & Node.js,” *International Journal of Research in Engineering and Science (IJRES) ISSN*, vol. 9, no. 7, pp. 29–37, 2021, [Online]. Available: [www.ijres.org](http://www.ijres.org)
- [27] L. R. Abbade, M. A. A. da Cruz, J. J. P. C. Rodrigues, P. Lorenz, R. A. L. Rabelo, and J. Al-Muhtadi, “Performance comparison of programming languages for Internet of Things middleware,” *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 12, Dec. 2020, doi: 10.1002/ett.3891.
- [28] R. S. Malik, J. Patra, and M. Pradel, “NL2Type: Inferring JavaScript Function Types from Natural Language Information,” in *Proceedings - International Conference on Software Engineering*, May 2019, vol. 2019-May, pp. 304–315. doi: 10.1109/ICSE.2019.00045.
- [29] C. Rong, W. Bin, and C. Yang, “Design of heterogeneous Data SQL access scheme based on postgresQL,” in *2019 IEEE 3rd International Conference on Electronic Information Technology and Computer Engineering, EITCE 2019*, Oct. 2019, pp. 1520–1524. doi: 10.1109/EITCE47263.2019.9094863.
- [30] C. Asiminidis, G. Kokkonis, and S. Kontogiannis, “Database Systems Performance Evaluation for IoT Applications,” *International Journal of Database Management Systems*, vol. 10, no. 06, pp. 01–14, Dec. 2018, doi: 10.5121/ijdms.2018.10601.
- [31] Z. Helmi, Thamrin, and S. Anita, “KUALITAS SUNGAI PETAPAHAN KECAMATAN GUNUNG TOAR KABUPATEN KUANTAN SINGINGI,” *Pusat Penelitian Lingkungan Hidup Universitas Riau*, pp. 46–58, 2013.
- [32] B. Suntaranont, S. Aramkul, M. Kaewmoracharoen, and P. Champrasert, “Water irrigation decision support system for practical weir adjustment using artificial intelligence and machine learning techniques,” *Sustainability (Switzerland)*, vol. 12, no. 5, Mar. 2020, doi: 10.3390/su12051763.