

Perancangan Node Sensor Nirkabel *Bluetooth Low Energy* Bertenaga Baterai menggunakan ESP32 untuk Aplikasi Pertanian Cerdas

Design of Battery-Powered Bluetooth Low Energy Wireless Sensor Node using ESP32 for Smart-Agriculture Purpose

Adi Pandu Wirawan, Happy Nugroho

Universitas Mulawarman, Jl. Sambaliung No. 09 Kampus Gunung Kelua Samarinda Kalimantan Timur

Email : adipandu@ft.unmul.ac.id

Abstrak – Peningkatan efektivitas dan efisiensi pertanian dapat dilakukan dengan penerapan teknologi sensor, otomatisasi, *monitoring* dan kendali jarak jauh melalui Jaringan Sensor Nirkabel (JSN). Salah satu teknologi komunikasi yang digunakan adalah *Bluetooth Low Energy* (BLE). Tersedianya *System on Chip* (SoC) ESP32 memudahkan implementasi BLE pada perangkat JSN. Penelitian ini bertujuan untuk menghasilkan prototipe node sensor berbasis BLE bertenaga baterai dan mengetahui kinerja prototipe tersebut terutama dalam hal pembacaan level baterai, jarak jangkauan dan akurasi pembacaan sensor agar dapat diimplementasikan pada JSN untuk aplikasi pertanian cerdas. Pada penelitian ini telah dihasilkan prototipe node sensor nirkabel BLE bertenaga baterai yang dapat mengirimkan data suhu, kelembapan dan informasi level baterai. Data-data yang dikirimkan telah dirancang dengan mengacu pada standar dan spesifikasi *Bluetooth Generic Attribute Profile* (GATT). Dalam pengujian menggunakan aplikasi *nRF Connect*, didapatkan hasil bahwa data yang dikirimkan telah sesuai dengan *attribute* standar GATT. Telah dilakukan pula pengujian pembacaan sensor suhu dan kelembapan dengan nilai *error* berturut-turut sebesar 1% dan 2,2%. Pengujian juga dilakukan terhadap pembacaan level baterai dan didapatkan nilai *error* sebesar 2,45%. Melalui pengujian jarak komunikasi antar node, node sensor mampu mengirimkan data dengan baik hingga jarak 10 meter. Dengan hasil tersebut maka dapat disimpulkan bahwa prototipe node sensor BLE bertenaga baterai yang dihasilkan dapat digunakan untuk aplikasi pertanian cerdas. Penelitian ini juga turut berkontribusi pada perkembangan ilmu pengetahuan khususnya dalam pemanfaatan BLE pada sistem pertanian cerdas yang masih jarang digunakan.

Kata kunci : JSN, BLE, ESP32, GATT, pertanian

Abstract - Increasing the effectiveness and efficiency of agriculture can be done by applying sensor technology, automation, monitoring, and remote control through the Wireless Sensor Network (WSN). One of the technologies used is *Bluetooth Low Energy* (BLE). The presence of the ESP32's *System on Chip* (SoC) makes it possible to implement BLE on WSN devices. This study aims to produce a prototype battery-powered BLE-based sensor node and determine the performance of the prototype, especially in terms of battery level readings, range, and accuracy of sensor readings so that it can be implemented in WSN for smart agricultural purposes. In this research, a battery-powered BLE sensor node prototype has been created. The node can transmit temperature, humidity, and battery level information. The data sent has been designed based on the *Bluetooth Generic Attribute Profile* (GATT) standards and specifications. *nRF Connect*'s application shows that the data sent by the designed node complies with the GATT attributes standard. Testing of temperature and humidity sensor readings has also been carried out with 1% and 2.2% errors respectively. Tests were also carried out on battery-level readings and a 2.45% error was obtained. Through distance testing between nodes, the node can transmit data properly up to 10 meters. With these results, it can be concluded that the resulting battery-powered BLE sensor node prototype can be used for smart agricultural purposes. This research also contributes to the development of science, especially in the use of BLE in a smart agriculture system which are rarely used.

Keywords : WSN, BLE, ESP32, GATT, agriculture

I. PENDAHULUAN

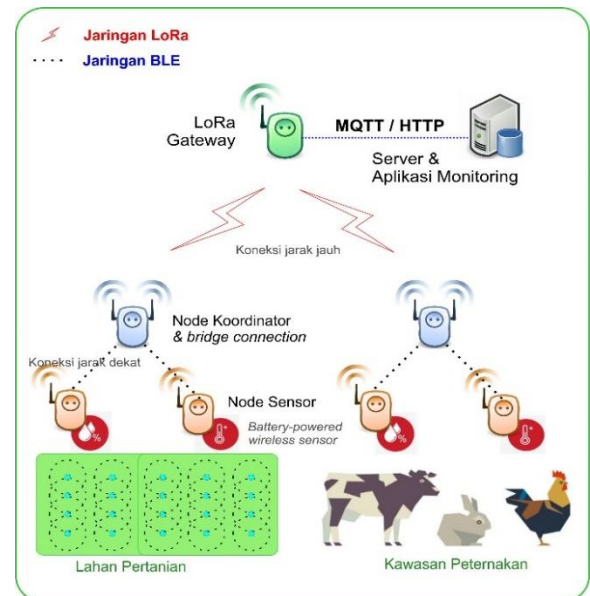
Pertanian cerdas atau *smart-agriculture* dapat dipahami sebagai sebuah konsep sistem pertanian modern yang menggunakan teknologi terbaru untuk menunjang produktivitas hasil pertanian yang maksimal [1]. Implementasi pertanian cerdas dapat dilakukan dengan penerapan teknologi sensor, otomatisasi, *monitoring* dan kendali jarak jauh melalui sistem Jaringan Sensor Nirkabel (JSN) ataupun jaringan yang lebih kompleks seperti *Internet of Things* (IoT).

Istilah JSN dapat didefinisikan sebagai sebuah jaringan yang menghubungkan perangkat-perangkat kecil yang disebut node sensor, tersebar dan mengambil sekaligus mentransmisikan data objek yang dipantau melalui jalur komunikasi nirkabel [2]. JSN memungkinkan terjadinya komunikasi data dari satu node sensor ke node lain tanpa adanya jaringan internet sehingga JSN cocok untuk digunakan pada wilayah yang memiliki konektivitas dengan sumber daya yang terbatas.

Pilihan teknologi komunikasi nirkabel pada JSN juga beraneka ragam mulai dari WiFi [3], *Long Range* (LoRa) [4], *Bluetooth standard* [5] hingga *Bluetooth Low Energy* (BLE) [6] dengan karakteristik dan keunggulannya masing-masing. Salah satu teknologi komunikasi yang memiliki keunggulan dalam hal konsumsi daya adalah *Bluetooth Low Energy* (BLE). BLE merupakan versi *improvement* dari *Bluetooth standard* dalam hal penghematan energi yang sering digunakan untuk aplikasi JSN maupun IoT dalam jarak dekat. Teknologi BLE menjadi yang paling unggul dari sisi *lifetime*, meskipun *lifetime* mengalami penurunan ketika ukuran data yang ditransmisikan semakin besar, namun dibanding dengan teknologi lain seperti WiFi 802.11 maupun LoRa, BLE tetap menjadi yang paling unggul dalam hal *lifetime* [7].

Teknologi BLE mengkonsumsi daya sangat kecil sehingga sangat cocok untuk diterapkan pada area pertanian atau peternakan yang wilayahnya jauh dari infrastruktur listrik yang memadai. Disisi lain BLE juga memiliki kekurangan yaitu terkait dengan jarak jangkauan komunikasinya yang lebih pendek jika dibandingkan dengan WiFi atau LoRa. Beberapa studi telah dilakukan untuk mengatasi kekurangan tersebut dengan tetap mendapatkan manfaat dari kelebihan BLE salah satunya adalah dengan *dual radio network* menggunakan LoRa untuk meningkatkan jarak jangkauan komunikasi [8-9]. Penggunaan *dual radio network* seperti yang

diilustrasikan pada **Gambar 1** menunjukkan penggunaan node sensor BLE memungkinkan untuk dapat dikembangkan lebih jauh terutama dalam implementasinya pada aplikasi pertanian cerdas.



Gambar 1. Arsitektur Dual Radio Network BLE dan LoRa

Hadirnya *System on Chip* (SoC) yang mendukung operasi protokol BLE pada perangkat keras seperti SoC ESP32 yang banyak tersedia di pasaran dengan harga terjangkau menjadikan implementasi perangkat JSN semakin mudah untuk diimplementasikan. Pemanfaatan baterai sebagai sumber daya node sensor (*battery-powered sensor node*) yang berbasis SoC ESP32 juga memungkinkan untuk diimplementasikan karena spesifikasi kebutuhan daya SoC ESP32 yang relatif rendah.

Penelitian sebelumnya tentang implementasi teknologi BLE menggunakan ESP32 bertenaga baterai telah banyak dikembangkan mulai dari sistem pengunci pintu otomatis [10] hingga sistem pengamanan kendaraan [11]. Pada penelitian-penelitian tersebut belum ditemukan cara bagaimana masing-masing prototipe yang dihasilkan dapat memonitor level daya baterai yang dikonsumsi sehingga tidak dapat diketahui kapan sensor akan mati akibat kehabisan daya. Selain itu terdapat juga penelitian lain tentang implementasi ESP32 bertenaga baterai pada aplikasi pertanian [12]. Dalam penelitian tersebut teknologi komunikasi yang digunakan adalah WiFi dengan koneksi internet. Dalam kasus ini, implementasi pertanian cerdas cenderung hanya dapat diimplementasikan pada area

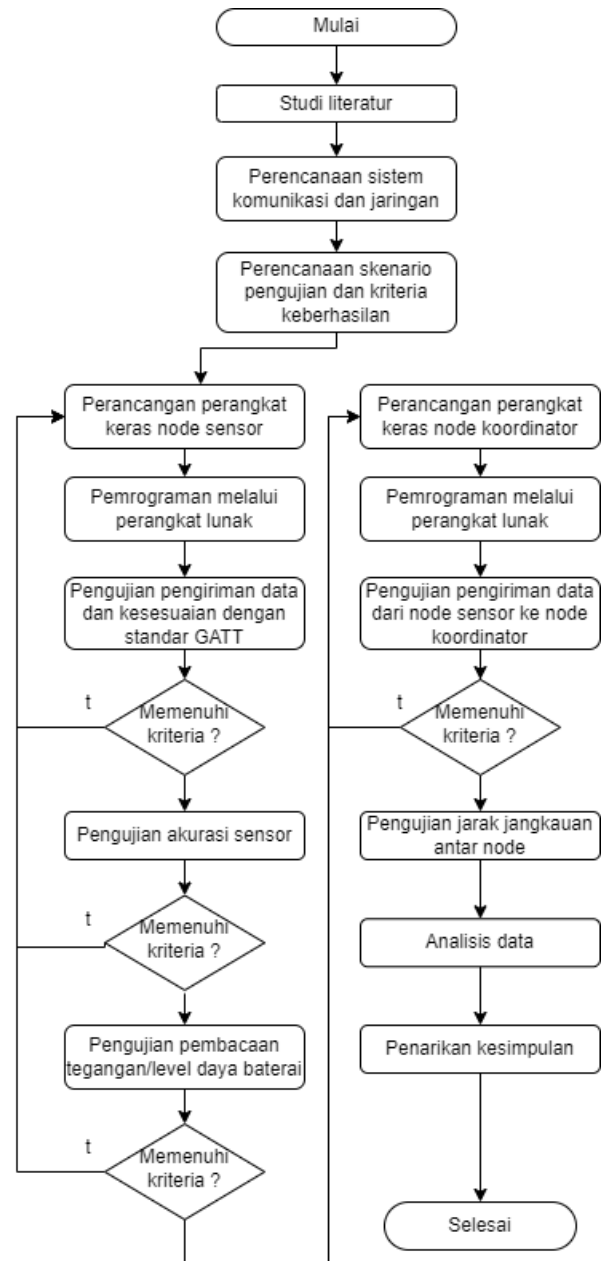
perkotaan (*urban farming*) atau area yang memiliki infrastruktur internet yang memadai. Sehingga kurang sesuai bila digunakan untuk area pertanian pedesaan (*rural farming*) dengan infrastruktur internet yang kurang memadai. Selain itu dibanding dengan BLE, penggunaan teknologi WiFi lebih banyak mengkonsumsi daya seperti yang sudah diuraikan sebelumnya.

Dengan adanya kekurangan-kekurangan pada penelitian terdahulu, penelitian tentang node sensor nirkabel BLE bertenaga baterai dipandang perlu untuk dilakukan. Penggunaan ESP32 sebagai node sensor bertenaga baterai cukup sesuai untuk diimplementasikan pada aplikasi pertanian cerdas yang wilayahnya cenderung jauh dari infrastruktur listrik. Perlu dilakukan investigasi lebih lanjut tentang cara pengiriman data, pembacaan sensor serta memonitor konsumsi daya melalui level baterai dari perangkat node sensor agar kekurangan daya dapat diprediksi sebelum node sensor mati. Atas dasar pertimbangan tersebut, perlu dilakukan penelitian untuk menghasilkan prototipe node sensor nirkabel berbasis BLE bertenaga baterai yang dapat digunakan pada aplikasi pertanian cerdas. Pengujian kinerja prototipe juga perlu dilakukan terutama dalam hal jarak jangkauan, akurasi pembacaan sensor dan pembacaan level baterai agar dapat diimplementasikan pada JSN untuk aplikasi pertanian cerdas.

Penelitian ini bertujuan untuk menghasilkan prototipe node sensor berbasis BLE bertenaga baterai dan mengetahui kinerja prototipe tersebut terutama dalam hal pembacaan level baterai, jarak jangkauan dan akurasi pembacaan sensor agar dapat diimplementasikan pada JSN untuk aplikasi pertanian cerdas.

II. METODOLOGI

Tiga rumusan penting yang perlu diketahui dalam penelitian ini adalah bagaimana merancang perangkat node sensor berbasis BLE agar dapat mengirimkan data *monitoring* dari sensor-sensor yang sering digunakan pada aplikasi pertanian cerdas, bagaimana melakukan *monitoring* konsumsi daya, serta bagaimana kinerja perangkat node sensor yang diusulkan terhadap jarak jangkauan komunikasinya. Untuk mendapatkan jawaban dari rumusan masalah tersebut, dilakukan tahapan-tahapan penelitian seperti ditampilkan pada **Gambar 2**.



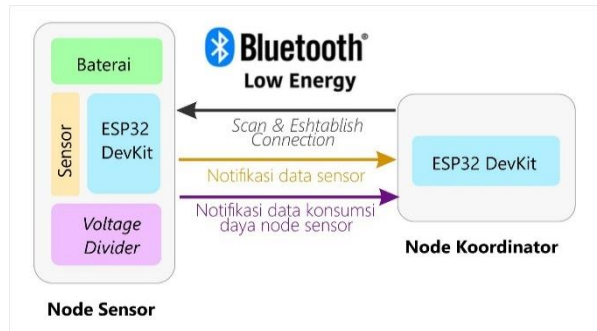
Gambar 2. Diagram Alir Metode Penelitian

A. Sistem Komunikasi dan Jaringan

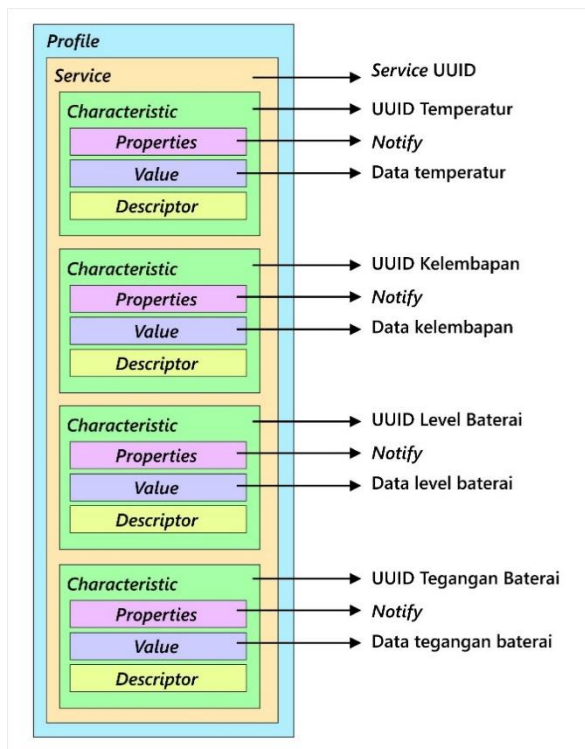
Untuk menghasilkan satu sistem komunikasi dengan konsep JSN perlu dilakukan perencanaan terkait topologi jaringan dan struktur data yang dikirimkan dari node pengirim ke node penerima. Pada penelitian ini, topologi jaringan yang digunakan adalah *point-to-point* dimana satu node sensor terkoneksi melalui protokol komunikasi BLE dengan satu node koordinator seperti ditampilkan pada **Gambar 3**.

Satu node sensor mengirimkan notifikasi berupa data suhu dan kelembapan yang didapat dari sensor serta data tegangan dari sumber baterai yang dipakai oleh node sensor itu sendiri untuk dikonversi menjadi informasi level baterai.

Pengiriman notifikasi tersebut dilakukan periodik setelah proses *handshake* koneksi antara kedua node berhasil dilakukan. Setiap node sensor dirancang memiliki *attribute* berupa *profile*, *service* dan *characteristic* mengacu pada standar *Bluetooth Generic Attribute Profile* (GATT). Struktur GATT yang dirancang pada node sensor ditampilkan pada **Gambar 4**.



Gambar 3. Komunikasi *Point-to-Point* Node Sensor



Gambar 4. Rancangan Struktur GATT

Satu node sensor dirancang terdiri dari satu *service* dengan 4 *characteristic* yang memuat data suhu, kelembapan, level baterai dan tegangan baterai. Masing-masing *characteristic* memiliki standar dan spesifikasi masing-masing dalam hal penggunaan *Universally Unique Identifier* (UUID), tipe data dan satuan yang mengacu pada standar spesifikasi BLE seperti ditampilkan pada **Tabel I**. Penyesuaian dengan standar tersebut dilakukan agar pengiriman data dari node sensor dapat secara universal diterjemahkan oleh

perangkat-perangkat penerima lain, tidak hanya node koordinator namun juga *smartphone* atau komputer tablet dengan menggunakan aplikasi pembaca data perangkat BLE.

Tabel I. Spesifikasi GATT Berdasarkan Data *Characteristic*

Data <i>Characteristic</i>	Deskripsi
Suhu	UUID: 0x2a6e Tipe data: uint16 Satuan: °C Resolusi: 0.01 Rentang: -273 s/d 328
Kelembapan	UUID: 0x2a6f Tipe data: uint16 Satuan: % Resolusi: 0.01 Rentang: 0 s/d 100
Level Baterai	UUID: 0x2a19 Tipe data: uint8 Satuan: % Rentang: 0 s/d 100
Tegangan Baterai	UUID: 6e400003 Tipe data: int UART via BLE

B. Skenario Pengujian

Terdapat beberapa tahap pengujian dan kriteria keberhasilan pengujian yang ditentukan untuk memastikan prototipe bekerja dengan baik dan sesuai dengan standar diantaranya:

- (1) Pengujian pengiriman data menggunakan aplikasi *nRF Connect* untuk melihat detail struktur GATT dari data yang dikirim node sensor dan menguji kesesuaian dengan spesifikasi *Bluetooth GATT*.
- (2) Pengujian akurasi sensor dilakukan menentukan dengan membandingkan nilai terukur pada sensor dan dengan nilai sebenarnya dari referensi alat ukur dengan besar *error* yang diharapkan dari pembacaan sensor adalah kurang dari 3% dimana nilai 3% didapatkan dari spesifikasi akurasi sensor AM2301 dalam pembacaan temperatur dan kelembapan.
- (3) Pengujian akurasi pembacaan tegangan dilakukan dengan membandingkan nilai tegangan terukur pada node dengan nilai sebenarnya dari voltmeter. Nilai pembacaan tegangan tergantung pada kinerja *Analog-to-Digital Converter* (ADC) yang digunakan. ADC ESP32 memiliki sifat non-linier terutama

pada nilai tegangan disekitar 0,1 volt, 3,1 volt dan 3,2 volt [13] sehingga sangat besar kemungkinan *error* pembacaan disekitar nilai tersebut. Dengan membandingkan 3 kemungkinan nilai *error* tersebut terhadap 33 data dalam rentang nilai pembacaan (0-3,3 volt *step size* 0,1) maka didapatkan angka pembacaan tegangan dengan toleransi sekitar 9%. Dengan demikian besar *error* yang diharapkan dari pembacaan tegangan adalah kurang dari 9%.

- (4) Pengujian *handshake connection* dan penerimaan notifikasi dari node sensor ke node koordinator dilakukan melalui serial monitor kedua node. Hasil pengujian dapat dikatakan sesuai kriteria jika koneksi BLE antar node berhasil dilakukan data yang diterima node koordinator sesuai dengan data yang dikirimkan oleh node sensor.
- (5) Pengujian jarak jangkauan antar node dilakukan untuk mengetahui seberapa jauh jangkauan komunikasi antar node.

C. Perancangan Node Sensor

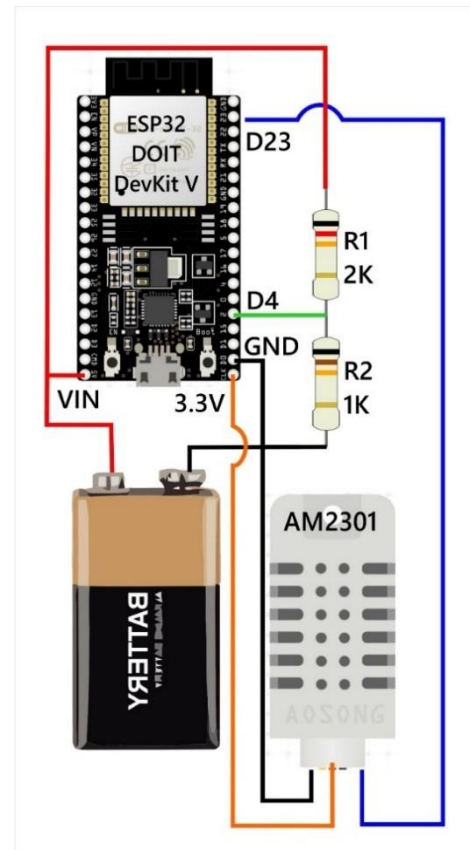
Rancangan perangkat keras node sensor dapat dilihat pada skematik yang ditunjukkan dalam **Gambar 5**. Diperlukan komponen penyusun yang terdiri dari sensor, baterai dan *development board* berbasis ESP32 seperti dirinci pada **Tabel II**. Rangkaian pembagi tegangan berupa dua buah resistor juga diperlukan untuk menyesuaikan kemampuan maksimum pembacaan tegangan ADC ESP32 agar tegangan baterai yang akan diukur tidak melebihi tegangan operasional dari ESP32 itu sendiri. Masing-masing komponen terhubung pada pin-pin pada *development board* ESP32 DOIT DevKit V1 dengan fungsi masing-masing pin dirinci pada **Tabel III**.

Tabel II. Daftar Komponen

No	Komponen	Jumlah
1.	ESP32 DOIT DevKit V1	1 Buah
2.	Sensor Suhu dan Kelembapan AM2301	1 Buah
3.	Baterai 9V	1 Buah
4.	Resistor 1K Ω	1 Buah
5.	Resistor 2K Ω	1 Buah

Tabel III. Konfigurasi Pin ESP32 DOIT DevKit V1

No	Pin	Fungsi
1.	VIN	Catu daya dari baterai
2.	3.3V	Catu daya untuk sensor
3.	D23	Input suhu dan kelembapan
4.	D4	Input ADC tegangan baterai



Gambar 5. Skematik Node Sensor

Pemrograman pada *development board* ESP32 dilakukan dengan menggunakan aplikasi *Arduino Web Editor* yang didapatkan dari layanan *Arduino Cloud*. Untuk membangun sistem komunikasi dengan protokol BLE pada ESP32 diperlukan inisiasi dan konfigurasi awal. Pada tahap inisiasi ini dilakukan pemanggilan *library*, konfigurasi pin, konfigurasi UUID dan tipe data kedalam program. Pemanggilan *library*, konfigurasi pin serta tipe data ditampilkan pada **Gambar 6**. Sedangkan konfigurasi UUID untuk setiap *attribute characteristic* yang direncanakan pada **Tabel I** ditampilkan pada **Gambar 7**.

```

Server.ino
1 #include <NimBLEDevice.h>
2 #include <DHT.h>
3 BLECharacteristic * pTempCharacteristic;
4 BLECharacteristic * pHumCharacteristic;
5 BLECharacteristic * pBatteryCharacteristic;
6 BLECharacteristic * pBatRawCharacteristic;
7 bool deviceConnected = false;
8 #define DHTPIN 23
9 const int pinBattery = 4;
10
11 #define DHTTYPE DHT21
12 DHT dht (DHTPIN, DHTTYPE);
13
14 uint16_t humidity;
15 uint16_t temperature;
16 uint8_t battLevel = 0;
17
18

```

Gambar 6. Inisiasi Library, Variabel dan Tipe Data


```

Server.ino
26 // UUID Service
27 #define SERVICE_UUID "6E400001-B5A3-F393-E0A9-E50E24DCCAE"
28 // UUID characteristic - Temperatur
29 #define DHTTEMP_CHAR_UUID "00002A6E-0000-1000-8000-00805F9B34FB"
30 // UUID characteristic - Kelembapan
31 #define DHTHUM_CHAR_UUID "00002A6F-0000-1000-8000-00805F9B34FB"
32 // UUID characteristic - Level Baterai
33 #define BATTERY_CHAR_UUID "00002A19-0000-1000-8000-00805F9B34FB"
34 // UUID characteristic - Tegangan Baterai (UART TX)
35 #define BATRAW_CHAR_UUID "6E400003-B5A3-F393-E0A9-E50E24DCCAE"
36

```

Gambar 7. Konfigurasi UUID

Indikator koneksi perlu digunakan pada node sensor agar pada saat pengujian dapat diketahui sukses atau tidaknya koneksi yang dilakukan, untuk itu digunakan fungsi koneksi dengan *output boolean* seperti yang ditampilkan pada Gambar 8.

```

Server.ino
39
40 class MyServerCallbacks: public BLEServerCallbacks {
41   void onConnect (NimBLEServer*pServer, ble_gap_conn_desc*desc) {
42     deviceConnected = true;
43     pServer->updateConnParams(desc->conn_handle, 24, 48, 10, 60);
44   };
45   void onDisconnect (NimBLEServer*pServer) {
46     deviceConnected = false;
47   }
48

```

Gambar 8. Fungsi Koneksi dengan Indikator Boolean

Setup node sensor diawali dengan memberikan nama "Node1 BLE-Agriculture" pada perangkat node sensor agar mudah dikenali dan membedakan dengan perangkat *Bluetooth* lainnya. Pada Gambar 9 ditampilkan pemberian nama node sensor sekaligus inisiasi pembuatan *service* dengan UUID yang sudah ditentukan sebelumnya. Langkah selanjutnya adalah membuat 4 jenis *characteristic* didalam *service* seperti yang ditampilkan pada Gambar 10. Setelah *service* dan *characteristic* dibuat dilakukan aktivasi *service* BLE dan menampilkan indikator awal yang menyatakan koneksi dengan node lain belum terhubung seperti ditampilkan pada Gambar 11. Untuk mendapatkan nilai suhu dan kelembapan dari sensor AM2301 dilakukan menggunakan fungsi *readHumidity()* dan *readTemperature()* yang terdapat pada *library DHT*. Selain itu juga dilakukan perkalian dengan nilai 100 untuk menyesuaikan dengan resolusi pembacaan nilai suhu dan kelembapan yang telah ditentukan pada spesifikasi *characteristic* GATT seperti ditunjukkan pada Gambar 12.

```

Server.ino
96 void setup() {
97   Serial.begin(9600);
98   dht.begin();
99
100   NimBLEDevice::init("Node1 BLE-Agriculture");
101   NimBLEServer *pServer = NimBLEDevice::createServer();
102   pServer->setCallbacks(new MyServerCallbacks());
103   NimBLEService *pService = pServer->createService(SERVICE_UUID);
104

```

Gambar 9. Pemberian Nama Perangkat dan Inisiasi Service

```

Server.ino
112 // Characteristic
113 pTempCharacteristic = pService->createCharacteristic(
114   DHTTEMP_CHAR_UUID,
115   NIMBLE_PROPERTY::NOTIFY
116 );
117 pHumCharacteristic = pService->createCharacteristic(
118   DHTHUM_CHAR_UUID,
119   NIMBLE_PROPERTY::NOTIFY
120 );
121 pBatteryCharacteristic = pService->createCharacteristic(
122   BATTERY_CHAR_UUID,
123   NIMBLE_PROPERTY::NOTIFY
124 );
125 pBatRawCharacteristic = pService->createCharacteristic(
126   BATRAW_CHAR_UUID,
127   NIMBLE_PROPERTY::NOTIFY
128 );

```

Gambar 10. Pembuatan Characteristic BLE

```

Server.ino
139
140 // Start service
141 pService->start();
142
143 // Menunggu koneksi Node Koordinator
144 pServer->getAdvertising() -> start();
145 Serial.println("Menunggu koneksi Node Koordinator ...");
146

```

Gambar 11. Menjalankan Service dan Menampilkan Status Koneksi

```

Server.ino
162
163 humidity = dht.readHumidity() * 100;
164 temperature = dht.readTemperature() * 100;
165

```

Gambar 12. Pembacaan Nilai Suhu dan Kelembapan

Pembacaan tegangan dan konversi level daya baterai perlu diimplementasikan kedalam program. Untuk membaca tegangan, dilakukan dengan menginisiasi nilai level minimum dan maksimum baterai serta mengetahui resolusi dari ADC yang digunakan. Berdasarkan dokumentasi ESP32, ADC memiliki maksimum resolusi sebesar 4095. Pada skematik Gambar 5, untuk mengetahui nilai tegangan input pada pin D4 yang dihasilkan dari R_2 (V_{R2}) dapat digunakan persamaan (1) dimana tegangan referensi (V_{ref}) yang digunakan pada ADC ini adalah 3.3V. Dengan mengetahui nilai-nilai resistor pada rangkaian pembagi tegangan dimana R_1 bernilai $2K\Omega$ dan R_2 bernilai $1K\Omega$, maka dapat diperoleh tegangan baterai (V_{in}) dari persamaan (2). Level daya baterai secara sederhana dapat diketahui dengan menentukan level tegangan minimum (V_{min}) dan level tegangan maksimum (V_{max}). Dalam hal ini level tegangan maksimum merupakan tegangan pada V_{R2} saat baterai dalam kondisi penuh yaitu 9 volt sedangkan level tegangan minimum ditentukan pada saat tegangan pada V_{R2} berada pada batas bawah tegangan operasional *development board* yang direkomendasikan yaitu 7 volt.

$$V_{R2} = \text{Data ADC} * \frac{V_{ref}}{\text{resolusi}} \quad (1)$$

$$V_{in} = V_{R2} * \frac{(R_1 + R_2)}{R_2} \quad (2)$$

Dengan menggunakan persamaan (2) dimana V_{max} sama dengan V_{R2} pada saat V_{in} bernilai 9 volt dan V_{min} sama dengan V_{R2} pada saat V_{in} bernilai 7 volt didapat V_{max} bernilai 3 volt dan V_{min} bernilai 2.33 volt. Dengan kondisi 100% merujuk pada nilai tegangan baterai 9 volt dan kondisi 0% merujuk pada nilai tegangan baterai 7 volt sehingga dapat diperoleh level baterai melalui persamaan (3). Implementasi persamaan (1), (2), dan (3) kedalam program ditampilkan pada **Gambar 13**.

$$Level = \frac{V_{R2} - V_{min}}{V_{max} - V_{min}} \quad (3)$$

```

Server.ino
144 void loop () {
145   if (deviceConnected) {
146     int VADC = 0;
147     float VREF = 3300;
148     float VIN = 0;
149     float VR2 = 0;
150     int resolution = 4095;
151     int R1 = 1950;
152     int R2 = 970;
153     float VMIN = 2330;
154     float VMAX = 3000;
155     float LEVEL = 0;
156     VADC = analogRead(pinBattery);
157     VR2 = (VADC*VREF)/resolution;
158     VIN = (VR2 * (R1+R2))/R2;
159     LEVEL = (VR2-VMIN)/(VMAX-VMIN);
160     if (LEVEL < 0) LEVEL = 0;
161     if (LEVEL > 1) LEVEL = 1;
162     battLevel = LEVEL*100;
163   }

```

Gambar 13. Pembacaan Tegangan dan Konversi Level Baterai

Untuk memastikan node sensor bekerja dengan baik, ditambahkan visualisasi pembacaan sensor dan tegangan serta level baterai melalui *serial monitor* seperti ditunjukkan pada **Gambar 14**. Pengiriman data masing-masing *characteristic* berupa notifikasi diimplementasikan kedalam program seperti ditampilkan pada **Gambar 15**.

```

Server.ino
178 Serial.print("Kelembapan:");
179 Serial.print(humidity/100);
180 Serial.println("%");
181 Serial.print("Temperatur:");
182 Serial.print(temperature/100);
183 Serial.println("C");
184 Serial.print("VADC:");
185 Serial.println(VADC);
186 Serial.print("VIN:");
187 Serial.print(VIN/1000);
188 Serial.println("V");
189 Serial.print("VR2:");
190 Serial.print(VR2/1000);
191 Serial.println("V");
192 Serial.print("Level Baterai:");
193 Serial.print(battLevel);
194 Serial.println("%");
195 Serial.print("UART Data String:");
196 Serial.print(vin);
197 Serial.println("");

```

Gambar 14. Menampilkan Nilai Pembacaan Sensor dan Tegangan

```

Server.ino
---
201 pTempCharacteristic->setValue(temperature);
202 pTempCharacteristic->notify();
203 pHumCharacteristic->setValue(humidity);
204 pHumCharacteristic->notify();
205 pBatteryCharacteristic->setValue(battLevel);
206 pBatteryCharacteristic->notify();
207 pBatRawCharacteristic->setValue(vin);
208 pBatRawCharacteristic->notify();
209
210 Serial.print ("---- Notifikasi dikirim ");
211 Serial.println ("----");
212 Serial.println ("");
213 }
214 delay (5000);
215 }

```

Gambar 15. Pengiriman Notifikasi Setiap 5 Detik

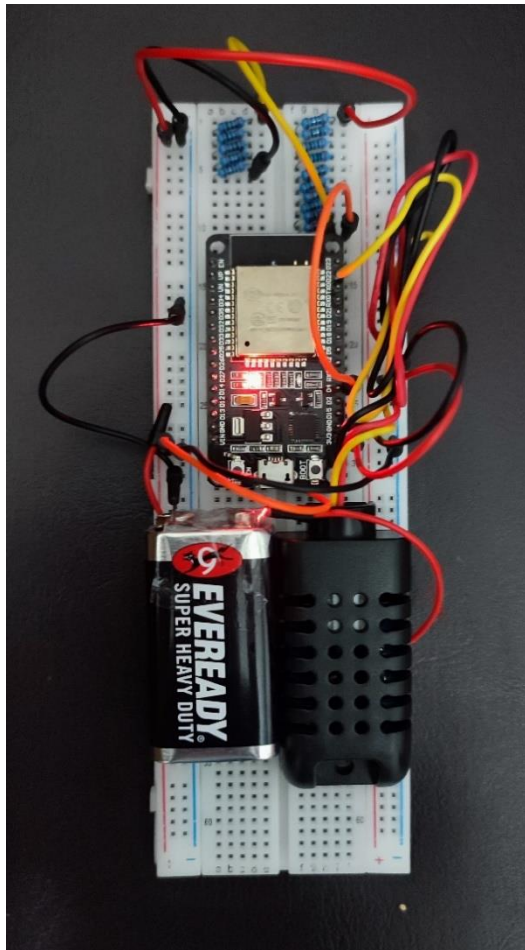
D. Perancangan Node Koordinator

Node koordinator berfungsi sebagai penerima notifikasi dari node sensor. Node koordinator dirancang menggunakan *development board* ESP32 DOIT DevKit V1. Dengan menggunakan *library NimBLEDevice*, ESP32 dapat difungsikan sebagai *client* yang melakukan koneksi otomatis ke *MAC Address* node sensor, kemudian melakukan koneksi ke *UUID service* dan membaca notifikasi dari setiap *characteristic*. Node koordinator terhubung dengan perangkat laptop untuk memonitor data serial dan memastikan data dari node sensor diterima dengan baik.

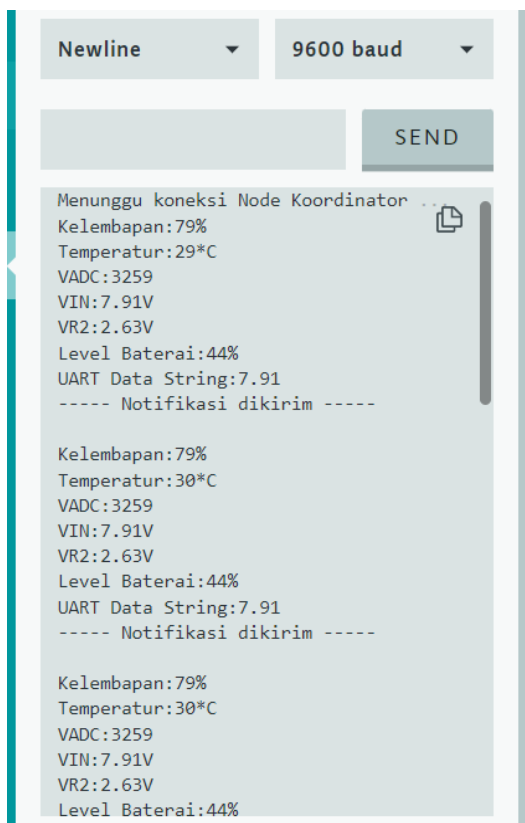
III. HASIL DAN PEMBAHASAN

Implementasi perangkat keras berupa prototipe node sensor sesuai dengan skematik yang telah dirancang dapat dilihat pada **Gambar 16**. Setelah rangkaian perangkat selesai dibuat, dilakukan pemrograman pada *development board* kemudian dilakukan *monitoring* pada *serial monitor* untuk memastikan program berjalan dengan baik. Hasil monitoring ditunjukkan pada **Gambar 17**. Pada hasil monitoring dapat dilihat bahwa node sensor dapat membaca dengan baik data suhu, kelembapan dan tegangan baterai beserta konversi levelnya.

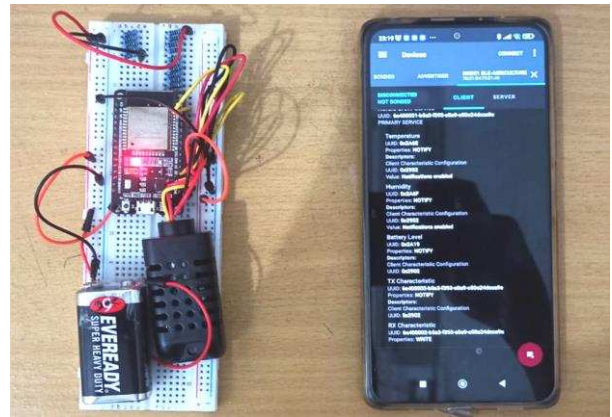
Uji coba pada sistem komunikasi juga dilakukan dengan menggunakan aplikasi *nRF Connect* pada *smartphone* seperti ditampilkan pada **Gambar 18**. Untuk memastikan perangkat BLE telah aktif memancarkan sinyal *Bluetooth*, dilakukan *scanning* pada aplikasi *nRF Connect* dan didapatkan hasil bahwa perangkat node sensor dapat terdeteksi dengan nama "Node1 BLE-Agriculture" seperti yang ditunjukkan pada **Gambar 19**. Koneksi yang dilakukan dari aplikasi ke node sensor berhasil memberikan informasi *characteristic* yang dikirimkan seperti ditunjukkan pada **Gambar 20**.



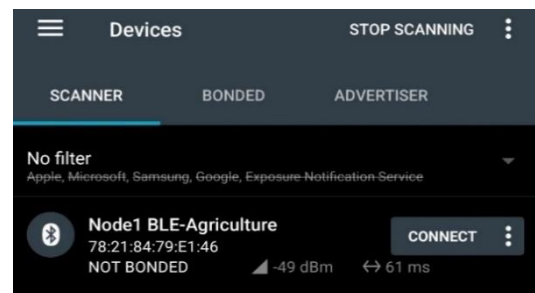
Gambar 16. Prototipe Node Sensor



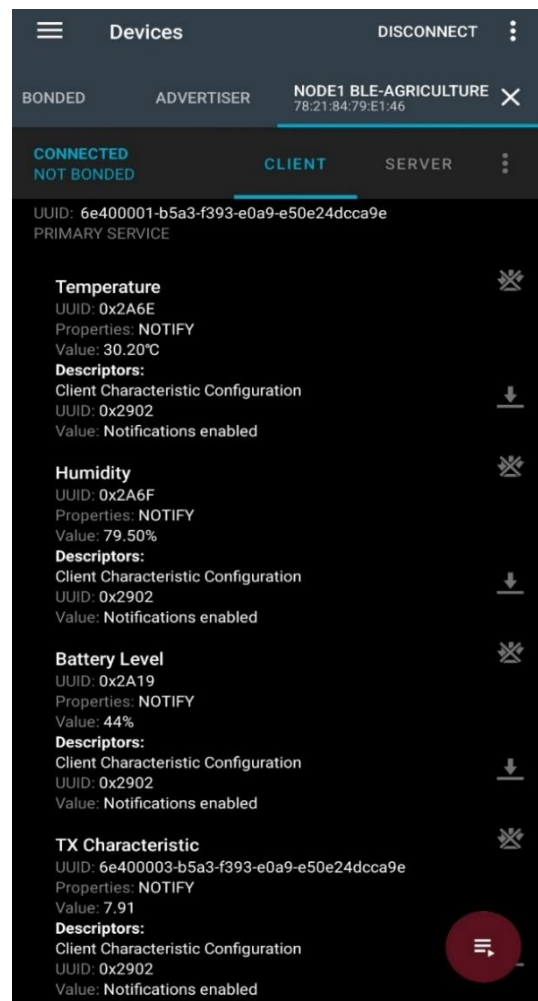
Gambar 17. Serial Monitor Node Sensor



Gambar 18. Pengujian Komunikasi Data menggunakan Smartphone



Gambar 19. Hasil Scanning Perangkat Bluetooth menggunakan Aplikasi nRF Connect



Gambar 20. Tampilan Real-time Data Characteristic Node Sensor

Masing-masing *characteristic* dapat teridentifikasi secara otomatis dengan nama-nama *characteristic* yaitu *Temperature*, *Humidity* dan *Battery Level*. Hasil ini menunjukkan bahwa spesifikasi *UUID characteristic* telah sesuai dengan standar GATT. Selain itu *value* setiap *characteristic* juga dapat menampilkan data sesuai dengan yang dikirimkan dan otomatis mengenali satuan pengukurannya, hal ini menunjukkan bahwa spesifikasi tipe data yang digunakan pada node sensor juga telah sesuai dengan spesifikasi GATT. Untuk *characteristic* tegangan baterai teridentifikasi dengan nama *TX Characteristic* tanpa satuan pada *value*, hal ini dikarenakan standar yang digunakan mengacu pada standar *Nordic UART Service* (NUS) yang memang diperuntukkan sebagai emulator UART/Serial Port melalui BLE sehingga dapat secara umum digunakan untuk acuan pengiriman data yang sifatnya umum atau spesifikasinya tidak terdapat pada standar GATT.

Hasil pengujian nilai pengukuran sensor suhu dan kelembapan berturut-turut ditampilkan pada **Tabel IV** dan **Tabel V**. Pengujian dilakukan dengan membandingkan antara nilai bacaan sensor dengan alat ukur *thermohygrometer* standar. Data sensor diambil dari *value characteristic* yang diterima *smartphone* dengan interval pengiriman data setiap 5 detik. Sebanyak 10 data sensor diambil sebagai sampel.

Tabel IV. Pengujian Sensor Suhu

Sampel	Sensor	Alat	Error	Error
	(°C)	Ukur (°C)	(%)	(°C)
1	29,8	29,5	1,0	0,3
2	29,8	29,6	0,7	0,2
3	29,7	29,6	0,3	0,1
4	29,8	29,6	0,7	0,2
5	29,8	29,7	0,3	0,1
6	30,1	29,7	1,3	0,4
7	30,1	29,7	1,3	0,4
8	30,2	29,7	1,7	0,5
9	30,1	29,7	1,3	0,4
10	30,2	29,7	1,7	0,5
Rata-rata Error			1	0,31

Pengambilan sampel data untuk masing-masing data suhu dan kelembapan dari sensor AM2301 menunjukkan kualitas hasil pembacaan yang baik karena rata-rata *error* hasil pembacaan sensor relatif kecil yaitu sebesar 1% atau 0,31°C untuk *error* pembacaan suhu serta 2,2% untuk *error* pembacaan kelembapan. Hasil ini sesuai dengan karakteristik sensor AM2301 seperti yang disebutkan pada penelitian lain tentang

penggunaan sensor AM2301 untuk keperluan node sensor [14] yang memiliki tingkat presisi sebesar $\pm 3\%$ untuk kelembapan dan $\pm 0.3^\circ\text{C}$ untuk suhu. Pada penelitian lainnya tentang karakteristik suhu dan kelembapan [15] juga memberikan kesimpulan yang menguatkan hasil pengujian dimana penelitian tersebut menyimpulkan bahwa akurasi sensor tidak melebihi $\pm 3,5\%$ untuk kelembapan dan $\pm 1^\circ\text{C}$ untuk suhu.

Tabel V. Pengujian Sensor Kelembapan

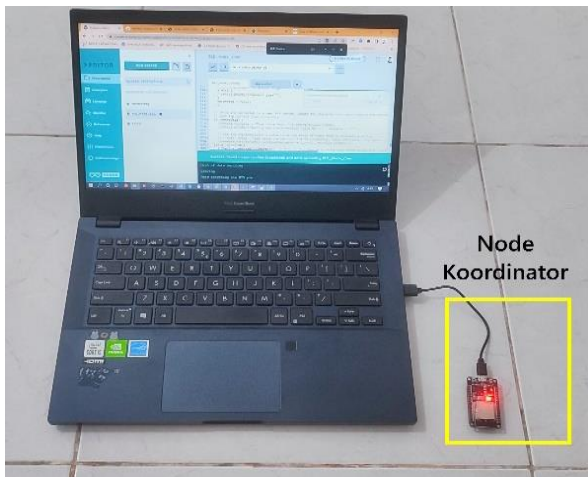
Sampel	Sensor	Alat	Error
	(%)	Ukur (%)	(%)
1	78,3	77,6	0,9
2	79,1	77,6	1,9
3	79,5	77,6	2,4
4	79,5	77,6	2,4
5	79,5	77,6	2,4
6	79,5	77,6	2,4
7	79,5	77,6	2,4
8	79,5	77,6	2,4
9	79,5	77,6	2,4
10	79,5	77,6	2,4
Rata-rata Error			2,2

Pengujian pembacaan tegangan baterai (V_{in}) dilakukan dengan membandingkan data pembacaan node sensor dengan tegangan terukur pada voltmeter. Untuk menguji pengukuran tegangan pada rentang 9 volt hingga 7 volt data sampel diambil pada rentang tersebut setiap penurunan tegangan sekitar 0,2 volt pada alat ukur. Hasil pengujian ditampilkan pada **Tabel VI**. Hasil pengujian menunjukkan *error* pembacaan tegangan berada dinilai rata-rata 2,45%. *Error* tersebut timbul dapat timbul selain akibat dari tidak liniernya ADC pada ESP32 [13] juga akibat dari akurasi nilai komponen pembagi tegangan seperti nilai resistor.

Tabel VI. Pengujian Tegangan Baterai

Sampel	Voltmeter	Pembacaan	Error
	(Volt)	Tegangan (Volt)	(%)
1	8,95	9,17	2,46
2	8,74	8,95	2,40
3	8,51	8,73	2,59
4	8,34	8,56	2,64
5	8,13	8,32	2,34
6	7,92	8,12	2,53
7	7,71	7,89	2,33
8	7,55	7,73	2,38
9	7,32	7,49	2,32
10	7,13	7,31	2,52
Rata-rata Error			2,45

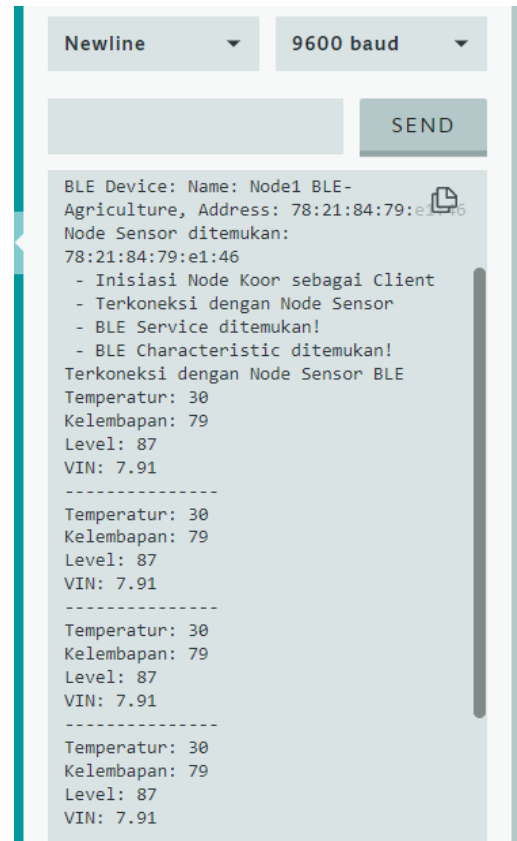
Setelah pengujian penerimaan data sensor melalui *smartphone* dilakukan, untuk mengetahui bahwa data sensor yang dikirimkan juga dapat diterima dengan baik oleh node koordinator maka dilakukan koneksi kembali antara node sensor dengan node koordinator yang terhubung dengan laptop seperti ditunjukkan pada **Gambar 21**.



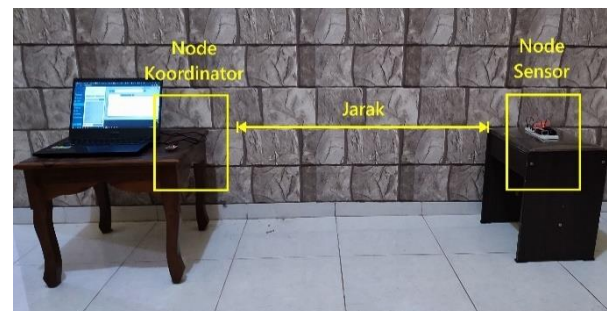
Gambar 21. Pengujian Node Koordinator

Koneksi diinisiasi oleh node koordinator dengan pertama kali melakukan *scanning* perangkat, kemudian mendeteksi node sensor dari MAC Address. Setelah koneksi berhasil dilakukan, node koordinator mencari *service* yang sesuai dengan service UUID dari node sensor. Setelah *service* ditemukan maka node koordinator akan mencari data *characteristic* dan melakukan *subscribe* pada notifikasi *characteristic* tersebut untuk mendapatkan pembaruan data secara periodik. Tahapan koneksi tersebut beserta pembacaan data yang sukses diterima node koordinator ditampilkan pada *serial monitor* seperti ditunjukkan pada **Gambar 22**. Hasil *monitoring* pada *serial monitor* menunjukkan bahwa data koneksi dengan node sensor berhasil dilakukan dan data yang diterima telah sesuai dengan yang data dikirimkan.

Pengujian selanjutnya dilakukan untuk mengetahui jarak jangkauan komunikasi antar node. Proses pengujian dilakukan dengan memisahkan node sensor dengan node koordinator pada jarak tertentu seperti ditunjukkan pada **Gambar 23**. Hasil pengujian jarak ditampilkan pada **Tabel VII**. Pada tabel tersebut dapat dilihat bahwa jarak maksimum antar node agar dapat terkoneksi dan mengirimkan data dengan stabil adalah pada nilai 10 meter.



Gambar 22. Serial Monitor Node Koordinator



Gambar 23. Pengujian Jarak Jangkauan Node BLE

Tabel VII. Pengujian Jarak Jangkauan

Jarak (m)	Status Koneksi
2	Tersambung
4	Tersambung
6	Tersambung
8	Tersambung
10	Tersambung
12	Terputus
14	Terputus

Jangkauan komunikasi BLE sangat dipengaruhi oleh frekuensi, daya pancar dan redaman yang timbul sepanjang sinyal berpropagasi dari pemancar hingga penerima. Pada hasil pengujian jarak didapatkan nilai maksimal jangkauan 10 meter. Hal ini disebabkan jalur propagasi yang dilalui bukan merupakan jalur ideal tanpa

penghalang atau *Non Line of Sight* (NLOS) selain itu daya pancar yang digunakan masih menggunakan daya pancar *default* yaitu 3 dBm. Berdasarkan spesifikasi teknis ESP32 memungkinkan untuk meningkatkan daya pancar BLE hingga 9 dBm untuk meningkatkan jarak jangkauan.

IV. KESIMPULAN

Pada penelitian ini telah dihasilkan prototipe node sensor nirkabel BLE bertenaga baterai yang dapat mengirimkan data suhu, kelembapan dan informasi level dan tegangan baterai. Dalam pengujian menggunakan aplikasi *nRF Connect*, didapatkan hasil bahwa data yang dikirimkan telah sesuai dengan spesifikasi GATT. Telah dilakukan pula pengujian pembacaan sensor suhu dan kelembapan dengan nilai *error* berturut-turut sebesar 1% dan 2,2%. Pengujian juga dilakukan terhadap pembacaan level baterai dan didapatkan nilai *error* sebesar 2,45%. Melalui pengujian jarak komunikasi antar node, node sensor mampu mengirimkan data dengan baik hingga jarak 10 meter. Dengan hasil pengujian kinerja node sensor bertenaga baterai tersebut dapat disimpulkan bahwa node sensor yang dirancang dapat digunakan untuk keperluan *monitoring* pada aplikasi pertanian hingga jarak 10 meter dari node koordinator. Dengan hasil tersebut maka dapat disimpulkan bahwa prototipe node sensor BLE bertenaga baterai yang dihasilkan dapat digunakan untuk aplikasi pertanian cerdas.

Aspek penghematan daya pada node sensor masih memungkinkan untuk dikembangkan lebih jauh dengan memanfaatkan kemampuan *deep-sleep* pada SoC ESP32 ataupun dengan menggunakan SoC lain yang memiliki konsumsi daya operasional yang lebih rendah. Selain itu peningkatan jarak jangkauan juga memungkinkan untuk dikembangkan lebih lanjut dengan mengkombinasikan teknologi BLE yang digunakan pada penelitian ini dengan teknologi lain yang memiliki keunggulan dari sisi jarak jangkauan.

DAFTAR PUSTAKA

- [1] T. Hidayat, "Internet of Things Smart Agriculture on ZigBee: A Systematic Review," *InComTech: Jurnal Telekomunikasi dan Komputer*, vol. 8, no. 1, hlm. 75–86, 2018.
- [2] P. Rawat, K. D. Singh, H. Chaochi, dan J. M. Bonnin, "Wireless Sensor Networks: A Survey on Recent Developments and Potential Synergies," *The Journal of Supercomputing*, vol. 68, no. 1, hlm. 1–48, Apr. 2014.
- [3] A. A. Masriwilaga, T. A. Jabar, A. Subagja, S. Septiana, "Sistem Monitoring Peternakan Ayam Broiler Berbasis Internet of Things," *TELEKONTRAN*, vol. 7, no. 1, hlm. 1-13, 2019.
- [4] F. A. Zulafah, D. Dewantama, dan Siswoko, "Rancang Bangun Stasiun Cuaca Berbasis Wireless Sensor Network dengan LORA SX1278," *TESLA: Jurnal Teknik Elektro*, vol. 24, no. 2, hlm. 116-128, Okt. 2022.
- [5] F. A. Ramadhany dan E. Yundra, "Rancang Bangun Aplikasi Wireless Body Area Network untuk Motion Sensing Berbasis Android dengan Media Transmisi Bluetooth," *Jurnal Teknik Elektro*, vol. 9, no. 2, hlm. 393-399, Mei. 2020.
- [6] A. Muttaqin, R. A. Setyawan, dan Muslichin, "Optimasi Daya Baterai Menggunakan Bluetooth Low Energy pada Routing di Wireless Sensor Network," *Jurnal EECCIS*, vol. 15, no. 2, hlm. 62–67, 2021.
- [7] É. Morin, M. Maman, R. Guizzetti, dan A. Duda, "Comparison of the Device Lifetime in Wireless Networks for the Internet of Things," pada *IEEE Access*, vol. 5, hlm. 7097-7114, 2017.
- [8] C. M. S. Ferreira, R. A. R. Oliveira, dan J. S. Silva, "Low-Energy Smart Cities Network with LoRa and Bluetooth," *2019 7th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, Newark, CA, USA, 2019, hlm. 24-29.
- [9] E. D. Ayele, K. Das, N. Meratnia, dan P. J. M. Havinga, "Leveraging BLE and LoRa in IoT network for wildlife monitoring system (WMS)," *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, Singapore, 2018, hlm. 342-348.
- [10] A. Prafanto, E. Budiman, P. P. Widagdo, G. M. Putra, dan R. Wardhana, "Pendeteksi Kehadiran menggunakan ESP32 untuk Sistem Pengunci Pintu Otomatis," *Jurnal Teknologi Terapan (JTT)*, vol. 7, no. 1, hlm. 37-43, 2021.
- [11] D. J. Managas, I. A. Supriyono, dan H. Junianto, "Perancangan Sistem Pencegahan Pencurian Kendaraan Bermotor Berbasis ESP32 pada PT. Suwarna Dwipa Maju," *Technomedia Journal (TMJ)*, vol. 7, no.2, hlm. 216-230, 2022.
- [12] A. Thoriq, L. H. Pratopo, R. M. Sampurno, dan S. H. Shafiyullah, "Rancang Bangun Sistem Monitoring Suhu dan Kelembaban Tanah Berbasis Internet of Things," *Jurnal Keteknikan Pertanian (JTEP)*, vol. 10, no. 3, hlm. 268-280, 2022.
- [13] M. Chilcott dan N. Kjærgaard, "Low-cost Wireless Condition Monitoring System for an Ultracold Atom Machine," *Internet of Things*, vol. 13, 2021.
- [14] F. Chen, L. Qin, X. Li, G. Wu, dan C. Shi, "Design and Implementation of ZigBee Wireless Sensor and Control Network System in Greenhouse," *Proceedings of the 36th Chinese Control Conference*, Dalian, China, 2017, hlm. 8982-8986.
- [15] A. V. Midyk dan O. Lysa, "Studies of Temperature-Humidity Characteristics of The Greenhouse Cyber-Physical System," *Measuring Equipment and Metrology*, vol. 81, no. 2, hlm. 7-12, 2020.