

PERANCANGAN KOMUNIKASI DATA ANTAR SISTEM INFORMASI MENGGUNAKAN TRANSMISI KONTROL PROTOKOL (TCP) PADA IOT SMART FARMING

Joko Triyono^{1*}, Elroy Abram Anugrahta Sitepu², Katherina Irene Dhamayanti³

^{1,2,3}Universitas AKPRIND Indonesia, Jl. Kalisahak Komplek Balapan Yogyakarta 0274-563029

Keywords:

ESP32, IoT, Local Area Network (LAN), Smart Farming, TCP Protocol

Correspondent Email:

jack@akprind.ac.id

Abstrak. IOT Smart Farming semakin hari menjadi semakin kompleks, dari yang hanya iot saja sampai bergerak ke skala yang lebih besar dan mengarah ke otomatisasi proses. Beberapa studi sudah dilakukan, baik itu dalam skala online maupun offline. Pada kenyataannya tidak semua proses IOT bisa diselesaikan semuanya dengan berbasis internet, banyak hal yang sebetulnya bisa diselesaikan secara *Local Area Network (LAN)*, Protokol TCP menjadi salah satu dari sekian banyak protokol yang bekerja untuk mengirimkan dan menerima data, Penelitian ini bertujuan untuk menerapkan protokol TCP dalam komunikasi data pada IOT secara local area network dengan multi platform. Dari hasil penelitian diperoleh bahwa protokol TCP ini bisa melakukan transfer data dari perangkat IOT dalam hal ini ESP32 ke Server berbasis Java dan juga melakukan proses dari Web Base PHP ke ESP32.



Copyright © [JITET](#) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. *IoT Smart Farming is becoming increasingly complex, evolving from basic IoT applications toward large-scale systems and process automation. Various studies have been conducted both online and offline. In practice, not all IoT processes require internet-based solutions; many can be effectively managed via a Local Area Network (LAN). The TCP protocol is one of many protocols used for data transmission and reception. This research aims to implement the TCP protocol for IoT data communication over a local area network across multiple platforms. The results demonstrate that the TCP protocol facilitates data transfer from IoT devices—specifically the ESP32—to a Java-based server, as well as executing processes from a PHP web interface to the ESP32.*

1. PENDAHULUAN

IOT Smart Farming saat ini menjadi topik hangat dan berkembang sangat cepat, baik itu skala kecil maupun skala besar. Dari yang hanya sekedar iot sederhana yang langsung ke obyek, iot yang terhubung ke cloud dan menyebar menjadi sebuah sistem informasi yang bisa membantu dalam pengelolaan smart farming secara efisien. Beberapa kajian masih mengadakan untuk selalu online atau terhubung internet, padahal pada kenyataannya tidak semua kondisi bisa langsung ke internet, cukup dengan

local area network (LAN) sudah bisa menyelesaikan masalah. IOT Smart Farming lebih mengedepankan untuk meningkatkan efisiensi dan optimalisasi dalam bidang ini, Bagaimana mengatur penyiraman otomatis yang terjadwal, mengetahui kelembaban tanah, perubahan suhu, pemupukan serta pemberian nutrisi dan obat-obatan secara efektif. Menurut Rusli implementasi smart farming berbasis IOT dapat menyelesaikan masalah pertanian dengan solusi yang efisien dan layak digunakan. Pada saat ini, Internet of things, dengan waktu yang

realtime dan akurat, dapat memberikan perubahan pada rantai pasokan pertanian dan menyediakan teknologi yang membuat pasokan logistik pertanian lancar. Aplikasi Internet Of Things (IOT) meliputi fasilitas proses data pertumbuhan tanaman, manajemen digital dari produksi, pembagian data, user Interface dan service intelligent network serta decision-making. Konsep dari Internet Of Things terkait dengan network, yang menkoneksi segala sesuatu dengan internet melalui Radio [1]. Menurut hasil dari penelitian Asty Novfelia, prototype yang dirancang bekerja baik menghasilkan rekomendasi kelayakan lahan berbasis suhu, kelembapan, CO₂, dan curah hujan. Sistem menerapkan fuzzy Mamdani secara lengkap—fuzzifikasi, inferensi 81 aturan, dan defuzzifikasi—untuk mengklasifikasikan kondisi menjadi Layak, Cukup Layak, Kurang Layak, atau Tidak Layak. Uji Mei–Juli 2025 di Nagari Selayo Tanang Bukit Sileh menunjukkan rata-rata Layak dengan skor 87,57%, sehingga lingkungan sangat baik untuk budidaya cabai. Metode ini efektif menangani ketidakpastian data sensor [2]. Dari penelitian yang dilakukan oleh I Wayan Indraswari Chaudhury Sandil dan Jarot Sugiharto, M. T. mengenai pengembangan sistem lift barang otomatis yang mengintegrasikan PLC dan ESP32, dapat disimpulkan bahwa penerapan teknologi Internet of Things (IoT) dalam sistem ini berhasil meningkatkan efisiensi operasional dan responsivitas. Dengan menggantikan Arduino dengan ESP32, sistem mampu menawarkan kontrol dan pemantauan jarak jauh melalui aplikasi Blynk, yang memungkinkan pengguna untuk mengakses dan mengendalikan lift dari mana saja. Keberhasilan semua percobaan yang dilakukan menunjukkan bahwa sistem berfungsi dengan baik dan mampu mengatasi tantangan yang ada pada lift konvensional. Selain itu, hasil percobaan yang menunjukkan waktu respons yang konsisten meskipun terdapat faktor delay jaringan, menegaskan efektivitas sistem dalam pengiriman sinyal antar lantai. Dengan demikian, penelitian memberikan solusi ini tidak modern hanya untuk pengoperasian lift barang, tetapi juga membuka peluang untuk pengembangan lebih lanjut dalam penerapan teknologi otomasi dan IoT di berbagai sektor industri. [3]. Berdasarkan hasil penelitian Joko Triyono dkk dengan tema Konsep Komunikasi Data Antar Sistem Informasi

Menggunakan Transmisi Kontrol Protokol (TCP) dapat disimpulkan bahwa: (1) Konsep ini membuktikan bahwa komunikasi antar sistem informasi bisa dilakukan menggunakan transmisi protokol protokol (TCP) dengan baik. (2) Komunikasi ini langsung mengakses ke rdbms dari sistem informasi yang ada. (3) Model komunikasi ini bisa digunakan untuk mengelola sistem informasi yang sudah ada dan tersebar dalam berbagai versi dan bentuk, tanpa harus mengganti dan membangun sistem informasi baru untuk menghadapi kebutuhan dan tantangan baru. [4] dan juga penelitian lain oleh Joko Triyono dkk dengan tema Konsep Sistem Terdistribusi pada pelayanan terpadu kalurahan menggunakan TCP disimpulkan bahwa konsep terdistribusi bisa digunakan untuk menghubungkan sistem informasi yang tidak berada dalam satu jaringan (saling asing) dengan bantuan sebuah server TCP yang diletakkan pada area publik, Keamanan Data akan lebih terjamin karena client tidak bisa mengakses data tersebut selain itu sistem informasi juga tidak diletakkan di area publik. Dengan konsep ini terbukti bahwa informasi dapat disebar/atau diakses oleh pihak luar sehingga nilai manfaat data dan informasi akan lebih meningkat. Beberapa saran yang belum diuji dalam penelitian ini adalah seberapa besar kemampuan server TCP dalam melayani Client APPS. [5]

Tujuan penelitian ini adalah menerapkan komunikasi data antar sistem informasi menggunakan protokol TCP dalam pengembangan IOT Smart Farming, dimana media input berupa sensor akan mengirimkan data melalui modul ESP32 dimana ESP32 menjadi sebuah node Client TCP akan mengirimkan data ke Server TCP melalui jalur wifi menggunakan port, sedangkan Server TCP akan melakukan penyimpanan data input tersebut, melakukan komputasi dan hasilnya selain disimpan ke dalam data juga akan di kirimkan ke Client ESP32 sebagai output untuk menggerakkan device lain sesuai dengan SOP yang telah ditentukan, selain itu Server juga akan mengirimkan data ke Client ADMIN berbasis web PHP melalui port TCP juga. Selain itu, Client Admin bisa meminta ke server TCP untuk memberikan data rekaman menjadi sebuah informasi tertentu, juga bisa melakukan perintah kepada client ESP32 sebagai output untuk mematikan atau menghidupkan device secara

langsung. Dan semua proses ini dilakukan secara local area network (LAN).

2. TINJAUAN PUSTAKA

Beberapa buku yang membahas tentang TCP [6] Arsitektur dan Mekanisme Kerja TCP, Model OSI dan TCP/IP dimana TCP bekerja pada lapisan transport dalam model OSI dan berfungsi di atas lapisan Internet dalam model TCP/IP. Bagian ini akan membahas peran TCP dalam model tersebut. Proses Establishment dan Termination terdiri dari dua model yaitu (A) Three-Way Handshake

SYN: Klien mengirim segmen SYN untuk memulai koneksi.

SYN-ACK: Server merespons dengan segmen SYN-ACK.

ACK: Klien mengirim segmen ACK untuk menyelesaikan pembentukan koneksi.

(B) Termination

[1]FIN: Salah satu pihak mengirim segmen FIN.

[2]FIN-ACK: Pihak lain mengakui FIN dan mengirim segmen FIN sendiri.

[3]ACK: Koneksi ditutup setelah segmen ACK diterima.

Juga tentang Pengendalian Aliran dan Kesalahan meliputi:

- Sliding Window: Mengelola ukuran jendela pengiriman untuk kontrol aliran.
- Retransmission: Mengirim ulang segmen yang hilang atau rusak.
- Checksum: Memverifikasi integritas data.
- d.

Beberapa Keuntungan Penggunaan TCP [7] seperti **Reliabilitas** dimana TCP menyediakan pengiriman data yang andal dengan mekanisme pengendalian kesalahan dan retransmisi. **Kontrol Aliran dan Kepadatan** dimana Mekanisme kontrol aliran dan kepadatan mencegah overload jaringan dan mengatur pengiriman data sesuai kemampuan penerima. Serta **Urutan Data** dimana TCP menjamin bahwa data diterima dalam urutan yang benar, sehingga aplikasi dapat memproses data secara konsisten. Juga bagaimana bentuk tantangan dalam Implementasi TCP [8] seperti Latensi dan Overhead Proses three-way handshake dan retransmisi dapat menambah latensi dan overhead dalam komunikasi data. Juga Skalabilitas TCP mungkin menghadapi masalah skalabilitas dalam jaringan dengan jumlah pengguna yang sangat besar atau dalam lingkungan yang sangat

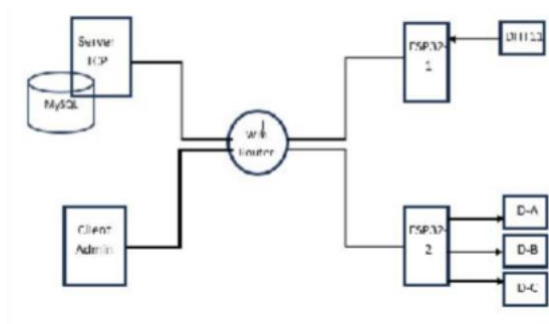
dinamis. Serta Keamanan Meskipun TCP menyediakan komunikasi yang andal, ia rentan terhadap serangan seperti TCP SYN Flooding yang dapat menyebabkan Denial of Service (DoS).

Koneksi Program Java dengan Database [9] di Buku Modul Praktikum dengan memanfaatkan JDBC. Juga tentang Java Network programming [10] serta beberapa referensi online [11], juga tentang TCP/IP Socket in Java [12]. Internet of Things (IoT) yang merupakan sekumpulan jaringan dan perangkat yang saling terhubung melalui internet, masing-masing memiliki identitas unik dan alamat IP. IoT memungkinkan perangkat untuk berkomunikasi satu sama lain, baik secara fisik maupun virtual, kapan saja [13]. IoT juga berfungsi untuk mengumpulkan dan menganalisis data dari berbagai sensor dan aktuator, yang kemudian dapat diakses melalui smartphone atau komputer secara nirkabel. Tujuan utama IoT adalah untuk mengaktifkan perangkat dan menjaga koneksi di mana saja dan kapan saja melalui jaringan. ESP32 adalah sebuah papan mikrokontroler yang sangat canggih dan serbaguna, dirancang oleh Espressif Systems. ESP32 terkenal karena kemampuannya yang lengkap, termasuk konektivitas nirkabel seperti Wi-Fi dan Bluetooth, sehingga sering digunakan untuk proyek Internet of Things (IoT). Mikrokontroler ini memungkinkan perangkat elektronik membaca input, memproses data, dan menghasilkan output sesuai dengan program yang ditanamkan di dalamnya. ESP32 merupakan mikrokontroler yang dapat terhubung dengan Wi-Fi, sehingga memungkinkan pengembangan perangkat berbasis IoT. ESP32 memiliki 48 pin, termasuk 18 pin ADC (12-bit), empat unit SPI, dan dua unit I2C. [14] ESP32 dilengkapi dengan beberapa fitur utama, seperti prosesor ganda (dual core), konektivitas nirkabel, dan banyak pin input/output (I/O), yang membuatnya ideal untuk berbagai aplikasi, mulai dari otomasi rumah hingga perangkat pintar. [15].

3. METODE PENELITIAN

Metode Penelitian dilakukan dalam skala laboratorium, dimana penelitian ini dilakukan di laboratorium komputer Jaringan dan Internet Jurusan Rekayasa Sistem Komputer Uni-

versitas AKPRIND Yogyakarta. Dengan membuat sebuah simulator yang akan mewakili beberapa objek penelitian. Gambar 3.1 menunjukkan design perencanaan simulasi.



Gambar 3.1. Design

Input sensor diperoleh dari sensor DHT11 berupa kelembaban dan suhu serta PH dalam penelitian ini diubah dengan nilai random yang akan dikirimkan oleh ESP321 ke Server TCP, melalui jalur Wifi, Server TCP akan merecord data masuk ke database, dan mengolah dengan algoritma yang telah ditentukan, serta mengirimkan ke Client Admin sebagai notifikasi, dan juga akan melakukan perintah untuk mengaktifkan device (DA, DB atau DC) pada ESP322, selain itu Server TCP juga akan mengaktifkan device tersebut sesuai schedule yang telah diatur di Database Server TCP secara otomatis. Client Admin selain menerima informasi dari Server TCP juga bisa mentrigger Device-device pada ESP322.

Dalam penelitian ini akan digunakan object SOP tanaman bawang merah sebagai data uji coba seperti diperlihatkan pada tabel 3.1.

Tabel 3.1 SOP Bawang Merah*

No	Keterangan	Syarat
1	Suhu	20°C – 30°C
2	pH Tanah	6 – 7
3	Kelembaban	60-70%
4	Penyiraman Rutin	Tiap Pagi jam 06.00
5	Pemupukan/Nutrisi	7hari 1x
6	Penyemprotan Insect n Nutrisi daun	3hari 1x

*Infrastruktur sudah tersedia

Proses Pengujian:

1. Data Sensor akan dikirimkan oleh ESP321 ke Server TCP tiap 5 menit 1 kali
2. Server TCP akan mengolah data sensor selama 30menit terakhir dan menjadi nilai rata2 dari masing-masing data sensor, untuk di bandingkan dengan SOP yang ada.

3. Data hasil olahan Server TCP yang LOW/HIGHT akan dikirimkan baik itu ke ESP322 maupun ADMIN sesuai dengan aturan yang berlaku.

4. Proses trigger ke Device dilakukan oleh Client Admin.

5. Device (DA) adalah device penyiraman, Device (DB) adalah device pemupukan, Device (DC) adalah device pemberian Nutrisi dan spray pendinginan.

6. Client Admin bisa melihat informasi/Data berbasis WEB ke Server TCP setiap dibutuhkan dan bisa melakukan aksi ke Client ESP322 secara langsung sesuai data yang ada.

Penyiapan Arsitektur Jaringan seperti terlihat pada tabel 3.2

Tabel 3.2 Arsitektur Jaringan

Node	IP Address	Keterangan
Wifi Router		SSID: routerwifi Password: router123
Server TCP	192.168.1.5	
Client Admin	192.168.1.6	
ESP321		Terhubung ke Wifi Router
ESP322		Terhubung ke Wifi Router

Penyiapan Database MySQL pada Server TCP, nama database iotsmartfarming, dengan tabel-tabel pendukung sebagai berikut

Tabel sop (id, keterangan, syarat, min, max, shedule, sudah, akan)

Tabel inputsensor (id, tgljam, asalclient, suhu, kelembaban)

Tabel outputdevice (id, idsop, tgljam, tujuan-client, tujuan device, waktu, trigger)

4. HASIL DAN PEMBAHASAN

Dari hasil penelitian dan pengujian laboratorium diperoleh hasil sebagai berikut.

4.1. Persiapan Infrastruktur Jaringan

Berdasarkan rancangan pada gambar 1 dan tabel 3.2 maka infrastruktur jaringan bisa diimplementasikan seperti pada tabel 4.1

Tabel 4.1 Implementasi Jaringan

Nama Alat/IP	Keterangan
Server TCP 192.168.1.5	Dihubungkan ke Wifi/Router melalui kabel UTP
Client ESP321	Dihubungkan ke Wifi Router meng-

192.168.1.3	gunakan Login SSID/Password yang ditanam di script ESP321
Client ESP322 192.168.1.4	Dihubungkan ke Wifi Router menggunakan Login SSID/Password yang ditanam di script ESP322
Client Admin/Web 192.168.1.6	Dihubungkan ke Wifi Router menggunakan Login SSID/Password
Wifi Router / Akses Point	Nama SSID: routerwifi Password: router123

4.2. Persiapan Database Server

Untuk menampung data transaksi yang terjadi dibutuhkan sebuah database, berikut adalah database dan tabel yang dibutuhkan.

Nama Database: iotsmartfarming

Tabel SOP, yang digunakan untuk menampung data SOP tanaman.

Tabel 4.2 Tabel sop

Nama Field	Type Data	Lebar	Keterangan
id	int		AutoIncrement PK
Keterangan	Varchar	50	
syarat	Int		
min	int		
max	int		
schedul	varchar	25	
sudah	datetime		
akan	datetime		

Tabel InputSensor, digunakan untuk mencatat semua data kiriman dari sensor di ESP321

Tabel 4.3 Tabel inputsensor

Nama Field	Type Data	Lebar	Keterangan
id	int		AutoIncrement PK
Tgljam	timestamp		
Asalclient	varchar	15	
Suhu	int	3	
Kelembaban	int	3	
ph	int	3	

Tabel OutputDevice, digunakan untuk mencatat hasil penghitungan data sensor dan data ini akan digunakan untuk mentrigger device ESP322 baik itu langsung atau melalui ADMIN.

(id, idsop, tgljam, tujuanclient, tujuan device, waktu, trigger)

Tabel 4.4 Tabel Outputdevice

Nama Field	Type Data	Lebar	Keterangan
id	int		AutoIncrement PK
idsop	int	11	index
tgljam	Timestamp		
tujuanclient	varchar	15	
tujuandevice	varchar	15	

Waktu	datetime		
trigger	varchar	15	

4.3. Client TCP ESP321

Client ini ditanam di mesin ESP32 dan bertugas untuk mendapatkan data dari sensor lalu mengirimkan ke Server TCP melalui protokol TCP dengan port 12345, sebelum mengirimkan maka script ini akan menghubungkan ESP32 melalui WIFI dengan SSID routerwifi dan password router123.

Berikut potongan Script ESP321

```

1. #include <WiFi.h>
2. // Konfigurasi WiFi & Server
3. const char* ssid = "routerwifi";
4. const char* password = "router123";
5. const char* host = "192.168.1.5";
6. const uint16_t port = 12345;
7. // Variabel Global yang akan diakses kedua Core
8. int globalsuhu = 0;
9. int globallembab = 0;
10. int globalPH = 0;
11. // Membuat Handle Mutex
12. SemaphoreHandle_t xMutex;
13. void setup() {
14. Serial.begin(115200);
15. xMutex = xSemaphoreCreateMutex();
16. // Koneksi WiFi
17. WiFi.begin(ssid, password);
18. while (WiFi.status() != WL_CONNECTED)
19. { delay(500); Serial.print("."); }
19. Serial.println("\nWiFi Terhubung!");
20. // 2. Membuat Task untuk TCP (Core 0)
21. xTaskCreatePinnedToCore(SendTCPTask,
22. "TCP_Task", 10000, NULL, 1, NULL, 0);
22. }
23. // --- LOOP UTAMA (Core 1): Membaca Sensor ---
24. void loop() {
25. // Simulasi baca sensor
26. int bsuhu = random(15, 50);
27. int bslembab = random(40, 90);
28. int bsPH = random(5, 8);
29. // Ambil kunci (Mutex) sebelum menulis data
30. if (xSemaphoreTake(xMutex, portMAX_DELAY)) {
31. globalsuhu = bsuhu; // Update data global
32. globallembab = bslembab;
33. globalPH = bsPH;
34. Serial.println("Core 1: Berhasil update data.");
35. // Lepas kunci kembali
36. xSemaphoreGive(xMutex);
37. }
38. delay(5000); // Baca sensor setiap 5 detik
39. }
40. // --- THREAD TCP (Core 0): Mengirim ke Server
41. // Java ---
41. void SendTCPTask(void *pvParameters) {
42. for(;;) {
43. WiFiClient client;
44. if (client.connect(host, port)) {
45. int ksuhu = 0; int klembab = 0; int kPH = 0;
46. // Ambil kunci sebelum membaca data agar tidak
    berubah saat dibaca

```

```

47. if (xSemaphoreTake(xMutex, portMAX_DELAY)) {
48.   ksuhu = globalsuhu;
49.   klembab = globallembab;
50.   kPH = globalPH;
51.   xSemaphoreGive(xMutex);
52. }
53. client.print("ESP321;" + String(ksuhu) + ";" +
  String(klembab) + ";" + String(kPH));
54. Serial.println("Core 0: Data terkirim ke Server Java:
  ");
55. client.stop();
56. }
57. // Kirim data setiap 5 detik
58. vTaskDelay(50000 / portTICK_PERIOD_MS);
59. }
60. }

```

Gambar 4.1 menunjukkan transaksi yang terjadi pada ESP321 saat membaca dan mengirimkan data sensor.

Gambar 4.1 Pembacaan dan Pengiriman data sensor dari ESP321

4.4. Client TCP ESP322

Client ESP322 di tanam pada mesin ESP32 bertindak menjadi client yang akan menjalankan device (DA, DB, dan DC) yang di trigger dari Client ESP321 maupun dari client ADMIN. Client ini akan menerima perintah/input stream berupa ON/OFF dari masing-masing device.

Berikut potongan Script ESP321

```

1. #include <WiFi.h>
2.
3. const char* ssid = "routerwifi";
4. const char* password = "router123";
5. const char* host = "192.168.1.5";
6. const uint16_t port = 12345;
7.
8. const int pinDA = 2; pinDB = 2; pinDC = 2; // LED
  bawaan ESP32
9.
10. void setup() {
11.   Serial.begin(115200);
12.   pinMode(pinDA, OUTPUT);
13.   pinMode(pinDB, OUTPUT);
14.   pinMode(pinDC, OUTPUT);

```

```

15.
16. WiFi.begin(ssid, password);
17. while (WiFi.status() != WL_CONNECTED)
  { delay(500); Serial.print("."); }
18. Serial.println("\nWiFi Terhubung!");
19. // Task untuk TCP (Kirim dan Terima)
20. xTaskCreatePinnedToCore(TCPCommunicationTask, "TCP_Task", 10000, NULL, 1, NULL, 0);
21. }
22. void loop() {
23.   // Logika lokal lainnya tetap berjalan di sini
24.   delay(1000);
25. }
26. // --- TASK KOMUNIKASI TCP ---
27. void TCPCommunicationTask(void *pvParameters)
  {
28.   for(;;) {
29.     WiFiClient client;
30.     if (client.connect(host, port)) {
31.       // 1. KIRIM DATA KE SERVER
32.       client.println("ESP322;STATUS:ACTIVE");
33.       // 2. TERIMA DATA DARI SERVER
34.       // Menunggu respon selama maksimal 2 detik
35.       unsigned long timeout = millis();
36.       while (client.available() == 0) {
37.         if (millis() - timeout > 2000) {
38.           break;
39.         }
40.       } while (client.available() > 0) {
41.         String request = client.readStringUntil('\n');
42.         request.trim(); // Hilangkan spasi atau \r
43.         Serial.println("Perintah dari Server: " + request);
44.         // bentuk data DC;ON;
45.         // Logika Case untuk Perintah
46.         if (request == "DA;ON") {
47.           digitalWrite(pinDA, HIGH);
48.           client.println("ACK:DA_NYALA");
49.         }
50.         else if (request == "DA;OFF") {
51.           digitalWrite(pinDA, LOW);
52.           client.println("ACK:DA_MATI");
53.         } else if (request == "DB;ON") {
54.           digitalWrite(pinDB, HIGH);
55.           client.println("ACK:DB_NYALA");
56.         }
57.         else if (request == "DB;OFF") {
58.           digitalWrite(pinDB, LOW);
59.           client.println("ACK:DB_MATI");
60.         }
61.         else if (request == "DC;ON") {
62.           digitalWrite(pinDC, HIGH);
63.           client.println("ACK:DC_NYALA");
64.         }
65.         else if (request == "DC;OFF") {
66.           digitalWrite(pinDC, LOW);
67.           client.println("ACK:DC_MATI");
68.         }
69.       }
70.       client.stop();
71.     }
72.     vTaskDelay(3000 / portTICK_PERIOD_MS); // Cek
  setiap 3 detik
73. } }

```

4.5. Server TCP

Server TCP menjadi bagian utama dari penelitian ini, selain menerima data, menyimpan data server ini juga akan melakukan komputasi untuk menghasilkan sebuah nilai yang akan disimpan ke database. Selain itu server ini juga mengelola data Client yang terhubung dan juga meneruskan data ke client lain. Dengan menggunakan teknik Thread maka proses akan berjalan simultan, tidak saling menunggu. Sedangkan untuk mengatur timer menggunakan

`scheduler.scheduleAtFixedRate(sendTask, 0, 31, TimeUnit.SECONDS);` dimana angka 31 menunjukkan bahwa unit ini akan dijalankan tiap 31 detik sekali, tanpa terganggu dengan proses yang lain.

Setiap ada data masuk, maka akan di check apakah pengirimnya ESP31, jika benar maka data tersebut akan disimpan di database.

Tiap 31 menit sekali maka akan dilakukan komputasi melalui view sql, dimana susunan view tersebut mulai dari mengambil 6 data sensor terbaru, menghitung rata, dan membandingkan dengan SOP dari tanaman, sehingga akan diperoleh data apakah LOW atau HIGH, jika kejadiannya LOW atau HIGH maka data hitungan akan disimpan ke dalam tabel outputdevice, dan akan dikirimkan baik itu ke ESP322 ataupun ke ADMIN.

Script Server.java yang handle semua komunikasi

```
1. import java.util.*;
2. import java.io.*;
3. import java.net.*;
4. import java.sql.Connection;
5. import java.sql.DriverManager;
6. import java.sql.PreparedStatement;
7. import java.sql.ResultSet;
8. import java.sql.SQLException;
9. import java.util.concurrent.Executors;
10. import java.util.concurrent.ScheduledExecutorService;
11. import java.util.concurrent.TimeUnit;
12. import java.util.Date;
13.
14. public class Server {
15.     private static final int PORT = 12345;
16.     private static Map<String, PrintWriter> clients =
new HashMap<>();
17.     // Konfigurasi Database
18.     private static final String URL = "jdbc:mysql://localhost:3306/iotmartfarming";
19.     private static final String USER = "iotmartfarming";
20.     private static final String PASS = "iotmartfarming";
21.     public static void main(String[] args) throws IOException {
22.         ServerSocket serverSocket = new ServerSocket(PORT);
23.         System.out.println("Server berjalan di port " +
PORT);
```

```
24.         while (true) {
25.             Socket socket = serverSocket.accept();
26.             new ClientHandler(socket).start();
27.         }
28.         static class ClientHandler extends Thread {
29.             private Socket socket;
30.             private String username;
31.             private BufferedReader in;
32.             private PrintWriter out;
33.             ScheduledExecutorService scheduler = Executors.newSingleThreadScheduledExecutor();
34.             public ClientHandler(Socket socket) {
35.                 this.socket = socket;
36.             }
37.             public void run() {
38.                 try {
39.                     in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
40.                     out = new PrintWriter(socket.getOutputStream(), true);
41.                     // Dapatkan username
42.                     username = in.readLine();
43.                     synchronized (clients) {
44.                         clients.put(username, out);
45.                     }
46.                     System.out.println(username + " terhubung.");
47.                     Runnable sendTask = () -> {
48.                         String data = " " + new Date().toString();
49.                         String target = "ESP322";
50.                         outputdevice(target);
51.                         System.out.println("Di kirim ke client: " + data);
52.                     };
53.                     scheduler.scheduleAtFixedRate(sendTask, 0, 26,
TimeUnit.SECONDS);
54.                     //
55.                     String message;
56.                     while ((message = in.readLine()) != null) {
57.                         if (message.startsWith("ESP321;")) {
58.                             //simpan ke tabel
59.                             simpanKeDatabase(message);
60.                         } else {
61.                             // Broadcast
62.                             for (Map.Entry<String, PrintWriter> entry : clients.entrySet()) {
63.                                 if (!entry.getKey().equals(username)) {
64.                                     entry.getValue().println("[ " + username + " ]: " +
message);
65.                                 }
66.                             } catch (IOException e) {
67.                                 System.out.println("Koneksi dengan " + username
+ " terputus.");
68.                             } finally {
69.                                 try {
70.                                     socket.close();
71.                                 } catch (IOException e) {}
72.                             }
73.                             synchronized (clients) {
74.                                 clients.remove(username);
75.                             }
76.                             // Format: ESP321;suhu;kelembaban;ph
77.                             private static void simpanKeDatabase(String data)
{
78.                                 String query = "INSERT INTO inputsensor (as-
alclient, suhu, kelembaban, ph) VALUES (?, ?, ?, ?)";
79.                                 String[] data1 = data.split(";");
```



```

127. }else{
128. String to322 = "DC;OFF";
129. targetesp322.println(to322);
130. }}
131. if(kete.equals("ph")) {
132. int turunph = resultSet.getInt("aksiturunph");
133. int naikph = resultSet.getInt("aksinaikph");
134. if(turunph>0){
135. String data = turunph+";ESP322;DB;ADMIN;
Turunkan PH karena rata2 ph "+resultSet.getFloat("phnya");
136. targetADMIN.println(data);
137. simpanaksi(data);
138. }
139. if(naikph>0){
140. String data = naikph+";ESP322;DB;ADMIN;
Naikkan PH karena rata2 ph "+resultSet.getFloat("phnya");
141. targetADMIN.println(data);
142. simpanaksi(data);
143. }}
144. if(kete.equals("kelembaban")) {
145. int turunl = resultSet.getInt("aksiturunlembab");
146. int naikl = resultSet.getInt("aksinaiklembab");
147. if(turunl>0){
148. String data = turunl+";ESP322;DB;ADMIN; Tur-
unkan Kelembaban karena rata2 ph "+resultSet.getFloat("lembab");
149. targetADMIN.println(data);
150. simpanaksi(data);
151. }
152. if(naikl>0){
153. String data = naikl+";ESP322;DB;ADMIN;
Naikkan Kelembaban karena rata2 ph "+resultSet.getFloat("lembab");
154. targetADMIN.println(data);
155. simpanaksi(data);
156. }} }}
157. connection.close();
158. } catch (SQLException e) {
159. System.out.println("Koneksi gagal: " + e.getMes-
sage());
160. }}}

```

```
>>> Data sensor 1/900 Wed Jan 14 23:34:19 WTR Z0G5
>>> Data sensor 30657 Wed Jan 14 23:34:54 WTD Z0G5
>>> Data sensor 30658 Wed Jan 14 23:34:59 WTR Z0G5
>>> Data sensor 4/439 Wed Jan 14 23:35:04 WTR Z0G5
>>> Data sensor 47487 Wed Jan 14 23:35:09 WTR Z0G5
$JMSQZD;0-03W331;Spray xarea sumc ratio 26.0 Wed Jan 14 23:35:11 WTB Z0G5
Data Refresh elapsed time PySQL output
El k: r= client: Wed Jan 15 08:15:11 WTR Z0G5
Data sensor 46657 Wed Jan 14 23:35:14 WTD Z0G5
>>> Data sensor 41654 Wed Jan 14 23:35:19 WTR Z0G5
>>> Data sensor 1/890 Wed Jan 14 23:35:24 WTB Z0G5
```

Create view ambilsuhu as


```
select `suhu` AS `suhu`,`kelembaban` AS `lembab`,`ph`
AS `ph` from `inputsensor` order by `id` desc limit 5
```

suhu	lembab	ph
47	86	6
41	80	6
44	66	7
47	46	7
47	49	7

Gambar 4.3 Select View ambilsuhu

Maka akan diperoleh 5 data sensor terbaru, selanjutnya melakukan komputasi rata-rata terhadap data tersebut.

Create view proses as

```
select avg(`ambilsuhu`.`suhu`) AS `suhunya`,avg(`ambilsuhu`.`lembab`) AS `lembab`,avg(`ambilsuhu`.`ph`) AS `phnya` from `iotsmartfarming`.`ambilsuhu`;
```

dari perintah tersebut diperoleh data komputasi terlihat pada gambar 4.4

suhunya	lembab	phnya
45.2000	65.4000	6.6000

Gambar 4.4. Select View Proses

Setelah diperoleh data rata-rata, lalu dilakukan perbandingan dengan data SOP menggunakan view

view checksuhu

```
select `id` AS `id`,`keterangan` AS `keterangan`,`syarat`
AS `syarat`,`minimal` AS `minimal`,`maksimal` AS `maksimal`,`terjadwal` AS `terjadwal`,`sudah` AS `sudah`,`akan` AS `akan`,`proses`.`suhunya` AS `suhunya`,`proses`.`lembab` AS `lembab`,`proses`.`phnya` AS `phnya`,
if(`proses`.`suhunya` > `maksimal`,`id`,0) AS `aksisuhu`,
if(`proses`.`lembab` < `minimal`,`id`,0) AS `aksinaiklembab`,
if(`proses`.`lembab` > `maksimal`,`id`,0) AS `aksitur-unlembab`,
if(`proses`.`phnya` < `minimal`,`id`,0) AS `aksinaikph`,
if(`proses`.`phnya` > `maksimal`,`id`,0) AS `aksitur-unph`,
case `id` when 3 then `suhu` when 4 then `ph` when 5 then `kelembaban` end AS `kete` from (`sop` join `proses`)
```

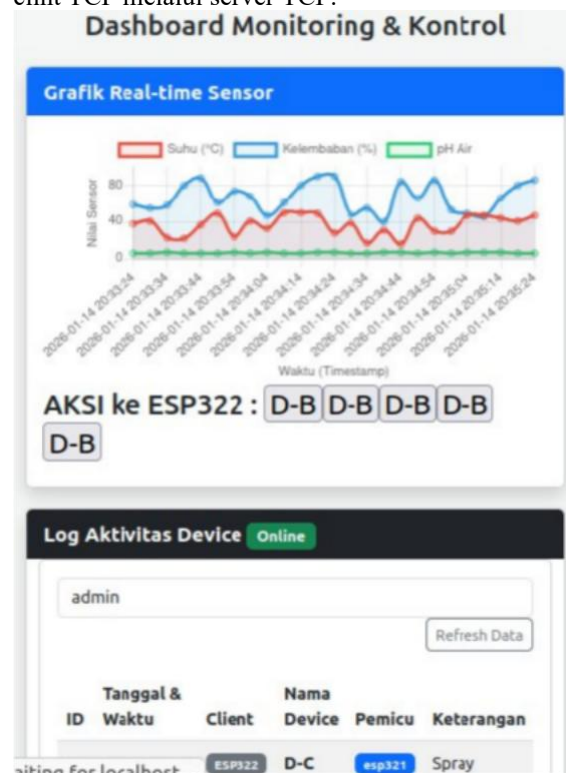
id	keterangan	syarat	minimal	maksimal	terjadwal	sudah	akan	suhunya	lembab	phnya	kete
3	Hentikan / Hentikan	0	0	0	0	0	0	45.2000	65.4000	6.6000	0
4	Berhenti / Berhenti	0	0	0	0	0	0	45.2000	65.4000	6.6000	0
5	Hentikan / Hentikan	0	0	0	0	0	0	45.2000	65.4000	6.6000	0

Gambar 4.5. Select View CheckSuhu

Dari proses tersebut akan diperoleh data akhir seperti terlihat pada gambar 4.5.

4.7. Client Admin / Web Base

Interface berbasis web digunakan untuk menampilkan dan monitoring semua data dan kejadian yang muncul, selain memonitor interface ini juga bisa bertindak untuk melakukan eksekusi atau perintah ke ESP322. Gambar 4.6 dashboard interface Web, Interface tersebut menampilkan grafik line real time data sensor dan tampilan data hasil komputasi, selain itu ada tombol untuk melakukan eksekusi ke ESP322, dimana dari tombol tersebut akan mengirimkan data atau parameter DC;ON, misalnya jika akan menghidupkan device DC, skrip berikut adalah script php yang akan menghubungkan php ke client TCP melalui server TCP.



Gambar 4.6 Interface Web

Script dari sisi Admin pada interface web (PHP) untuk memicu aksi pada ESP322,

```
1. <?php
2. $target = "ESP322";
3. $command = $_GET['q'];
4. $socket = fsockopen("192.168.1.5", 12345);
5. if ($socket) {
6. // Format: [ID_TARGET]:[PERINTAH]
7. fwrite($socket, "$target:$command\n");
8. echo fgets($socket); // Baca balasan: "Berhasil mengirim..."
9. fclose($socket);
10. }
11. header("Location: index.php");
12. ?>
```

Dari hasil pengujian terbukti bahwa transaksi pengiriman data menggunakan protokol TCP berhasil dilakukan, proses komputasi menggunakan view akan meminimalisir algoritma di sisi coding program karena dilakukan di area RDBMS, dalam pengujian ini juga membuktikan bahwa protokol TCP juga bisa bekerja dalam banyak aplikasi, dimana Client TCP berbasis ESP32 menggunakan C#, Server TCP menggunakan java dan interface web menggunakan PHP HTML. Dalam proses pengujian data juga tidak ditemukan kendala yang berarti saat ada proses yang bersamaan, dengan memanfaatkan teknik Thread, sehingga data tetap bisa berjalan tanpa harus menunggu antrian kapan harus dikerjakan. Fungsi timer menggunakan teknik `scheduler.scheduleAtFixedRate(sendTask, 0, 31, TimeUnit.SECONDS)`; juga ternyata efektif dalam melakukan penghitungan / timer.

5. KESIMPULAN

- Proses pengiriman data menggunakan TCP dalam penelitian ini berhasil dilakukan, baik dari sisi ESP32, Server TCP maupun dari Client ADMIN.
- Transaksional kedalam RDBMS bisa dilakukan dengan sempurna.
- Penggunaan View SQL secara bertingkat dalam melakukan komputasi cukup efektif dan meringankan dari sisi coding, karena semua algoritma dilakukan di sisi RDBMS.
- Protokol TCP bisa menjadi jembatan untuk melakukan komunikasi data antar aplikasi yang berbeda platform.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini. Khususnya Rektor Universitas Akprind Indonesia, Dekan Fakultas Sains dan Teknologi Informasi Universitas Akprind Indonesia, Ketua Program Studi Rekayasa Sistem Komputer serta tim Laboratorium Komputer Jaringan dan Internet yang telah memberikan support dalam penelitian ini.

DAFTAR PUSTAKA

- [1] S. J. Rusli, "IMPLEMENTASI KONSEP SMART FARMING BERBASIS IOT DAN MANFAATNYA," Jurnal Ilmu Teknik dan Komputer (JITKOM), vol. 5, no. Jurnal Ilmu Teknik Komputer, pp. 238-256, 1 Januari 2021.
- [2] A. Novfelia, B. P. Mandarin, E. K. Putra and A. Syahrani, "SISTEM MONITORING LINGKUNGAN BERBASIS IOT SEBAGAI REKOMENDASI WAKTU TANAM CABAI DI NAGARI SELAYO TANANG BUKIT SILEH," JITET (Jurnal Informatika dan Teknik Elektro Terapan), vol. 13, no. 3S1, pp. 860-866, 2025.
- [3] I. W. I. C. S. Sandi, "IMPLEMENTASI SISTEM KONTROL LIFT BARANG 3 LANTAI BERBASIS IOT," JITET (Jurnal Informatika dan Teknik Elektro Terapan), vol. 3, pp. 382-389, 2028.
- [4] J. Triyono, H. and R. Tefa, "Konsep Komunikasi Data Antar Sistem Informasi Menggunakan Transmisi Kontrol Protokol (TCP)," Konsep Komunikasi Data Antar Sistem Informasi Menggunakan Transmisi Kontrol Protokol (TCP), vol. 9(1), pp. 34-41, 2024.
- [5] J. Triyono, E. A. Anugraha Sitepu and A. Kurniawan, "Konsep Sistem Terdistribusi pada pelayanan terpadu kalurahan menggunakan TCP," in Prosiding Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB) 2025, Surakarta, 2025.
- [6] W. Steven, TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1994.
- [7] D. Comer, Internetworking with TCP/IP Volume one, Prentice Hall, 2000.
- [8] A. & W. D. Tanenbaum, Computer Networks, Prentice Hall, 2010.
- [9] A. Hamzah, E. Susanti and S. , MODUL PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK LANJUT, Yogyakarta: AKPRIND PRESS, 2018.
- [10] E. R. Harold, Java Network Programming Fourth Edition, -: O'Reilly Media; Fourth Edition (November 12, 2013), 2013.
- [11] "Java Documentation," 01 05 2024. [Online]. Available: <https://docs.oracle.com/javase/tutorial/networking/index.html>.
- [12] M. J. D. Kenneth L. Calvert, TCP/IP Sockets in Java: Practical Guide for Programmers, Morgan Kaufmann Publishers, 2002.
- [13] A. Winarno, W. and M. K. Nurcahyo, "Smart Home Remote Control System Prototype Using Internet of Things (IoT) Based ESP8266 Microcontroller," Journal of Applied Electrical Sains and Technology (beest), vol. 5, no. 2, pp. 29-33, 2023.
- [14] A. A, A. S. S.M. and B. D.F., "Implementasi Purwarupa Wireless Sensor Network Untuk Monitoring Dan Penyiraman Otomatis Pada Tanaman Mint Menggunakan Esp32 Berbasis IoT LoRa," J. Inform. dan Tek. Elektro Terap., vol. 12, no. 3, pp. 2113-2121, 2024.

- [15] J. J., M. F. and M, "Sistem Monitoring Dan Pengontrolan Elevator Berbasis Iot Dengan Sistem Pengendali Microcontroler Esp32 Pada Miniatur Elevator," JuTEkS (Jurnal Tek. Elektro dan Sains), vol. 9, no. 1, pp. 1-6, 2022.
- [16] N. N. M, H. Yuana and Z. Wulansari, "Mikrokontroler Esp 32 Sebagai Alat Monitoring Pintu Berbasis Web," JATI (Jurnal Mahasiswa Teknik Informatika), vol. 6, no. 2, pp. 767-772, 2022.