Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

June 2026. Vol. 5. No. 3; e-ISSN: 2808-4519

Predicting Chili Pepper Diseases using a Decision Tree in an Android-Based Internet of Things (IOT) Monitoring System

Novenda Putra Linarta Sitepu[1*], Relita Buaton[2], Magdalena Simanjuntak[3]

[1,2,3] Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Kaputama, Binjai
linartasitepunovendaputra@gmail.com [1], bbcbuaton@gmail.com[2], magdalena.simanjuntak84@gmail.com[3]*

Abstract

Chili pepper plants are susceptible to diseases caused by changes in the microclimate, making a data-driven monitoring system essential. This study designed an Internet of Things (IoT) system to monitor the microclimate and predict disease risks in chili pepper plants via an Android app. The system uses an ESP32 connected to a DHT22 sensor, a capacitive soil moisture sensor, a BH1750 sensor, a rain sensor, and a DS3231 RTC. Data on air temperature, air humidity, soil moisture, light intensity, and rainfall conditions are sent to the Firebase Realtime Database via WiFi in real-time. Predictions are made using the CART Decision Tree algorithm with low, medium, and high risk classifications. Test results show that the model achieved an accuracy of 95%, precision of 96%, recall of 95%, and an F1-score of 95%, with 19 out of 20 test data points correctly classified. This system helps farmers make cultivation decisions more quickly and objectively based on actual environmental conditions in chili farming fields.

***Keywords**: Android, CART, Internet of Things, Microclimate, Chili Plants.*

1. Introduction

Chili peppers (Capsicum annuum) are a strategic horticultural commodity in Indonesia that directly impacts food price stability and inflation. High demand is not always matched by stable production. One of the main factors affecting chili pepper productivity is the microclimate, which includes air temperature, relative humidity, soil moisture, light intensity, and rainfall. Imbalances in these parameters can increase the risk of growth disturbances and plant disease outbreaks.[1]

In agricultural practice, most farmers still rely on manual observations and field experience to assess plant conditions. This approach has limitations because it does not provide continuous, real-time environmental data. As a result, decisions regarding irrigation, fertilization, and disease control are often not based on objective data. This situation can lead to inefficient water use, reduced crop growth quality, and an increased risk of disease outbreaks due to uncontrolled environmental conditions.[2]

Advances in Internet of Things (IoT) technology offer opportunities to upgrade conventional agricultural systems to data-driven systems.[1] IoT enables the automatic collection of environmental data via sensors, which is then transmitted to a digital platform for real-time processing and display[3]. In the context of agriculture, this technology allows for the monitoring of microclimate conditions without spatial or temporal limitations, enabling faster and more accurate decision-making.[1], [3], [4]

In addition to IoT-based monitoring, artificial intelligence approaches are also increasingly being applied in agriculture. One commonly used method is the Decision Tree, specifically the Classification and Regression Tree (CART) algorithm, which can classify data based on patterns. This method offers the advantage of simple result interpretation and high accuracy with numerical data such as microclimate parameters. The integration of IoT and machine learning enables the system not only to monitor but also to predict the risk of plant diseases.[5]

Although various studies have developed IoT systems for smart agriculture, most still focus on environmental monitoring or irrigation automation. The integration of real-time microclimate monitoring, cloud-based data storage, mobile applications, and disease risk prediction into a single integrated system remains limited. Furthermore, implementing a system that is easily accessible via Android devices remains a challenge in the development of digital agriculture systems.[2], [4]

This study aims to design and implement an ESP32-based Internet of Things system capable of real-time microclimate monitoring for chili pepper plants. This system integrates a DHT22 sensor, a capacitive soil moisture sensor, a BH1750 sensor, and a rain sensor, and uses the Firebase Realtime Database as its data storage medium. Furthermore, the system is equipped with an Android app for data visualization and a CART Decision Tree model to predict the disease risk level of chili plants. The research results are expected to provide a more efficient, accurate, and accessible precision agriculture system solution.

2. Research Methodology

In this study, the methodology focuses on the design and development of an Internet of Things (IoT) system for microclimate monitoring, as well as the implementation of a Decision Tree (CART) algorithm to predict disease risk in chili plants, integrated with an Android application.

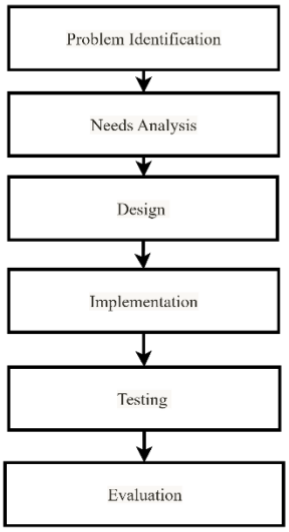


Fig. 1: Research Phases

2.1. System overview

This study developed an Internet of Things (IoT)-based microclimate monitoring system for chili plants that is integrated with an Android app and the Firebase Realtime Database. The system is designed to read environmental parameters in real time using sensors connected to an ESP32 microcontroller, then send the data to the cloud to be displayed on the user's app and processed using a CART Decision Tree model to predict the risk of plant diseases.

The system consists of three main components: a data acquisition device (sensors and ESP32), a cloud system (Firebase Realtime Database), and a mobile app (Android). The integration of these three components enables real-time monitoring of environmental conditions and disease risk prediction.

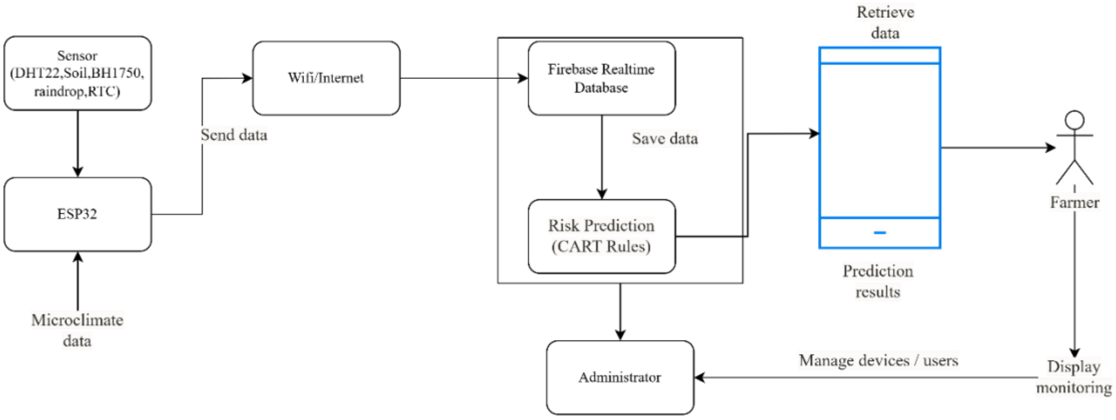


Fig. 2: Overall System Architecture (IoT–Cloud–Android Integration)

2.2. Hardware Configuration

The hardware used in this system consists of an ESP32 microcontroller as the control center, as well as several sensors for acquiring microclimate data for chili plants. The main components include:

1. ESP32 as the processing unit and Wi-Fi communication module[6]
2. DHT22 sensor for air temperature and humidity[7]

3. Capacitive soil moisture sensor for soil moisture content[8]
4. A BH1750 sensor for light intensity
5. A raindrop sensor for precipitation conditions
6. A DS3231 RTC for data timestamping
7. Driver circuitry and power supply

These sensors are integrated to generate environmental data used as input for the prediction system.

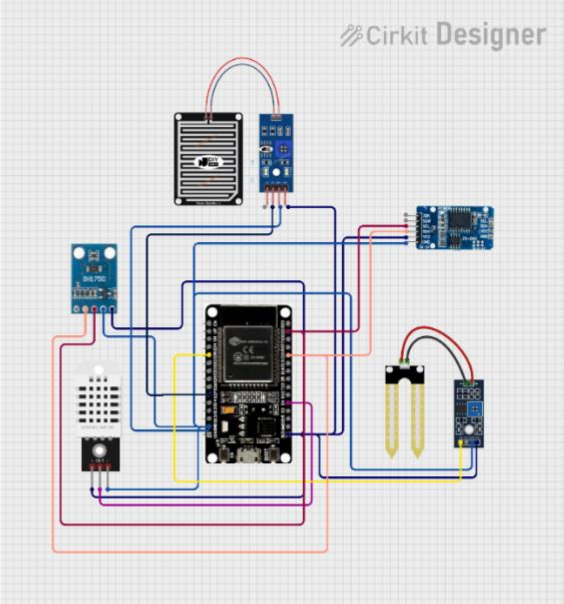


Fig. 3: Hardware System Design

2.3. Dataset and Feature Selection

The dataset in this study consists of microclimate data for chili plants, including air temperature, relative humidity, soil moisture, light intensity, and rainfall conditions. This data is used as features (inputs) in the classification model.

The output of the dataset is the level of disease risk for chili plants, categorized into three classes: low, moderate, and high.

Table 1: Dataset Features and Label

Feature	Description	Type
Air Temperature	Air temperature	Numeric
Air Humidity	Air humidity	Numeric
Soil Moisture	Soil moisture	Numeric
Light Intensity	Light intensity	Numeric
Rain Condition	Rain conditions	Categorical
Disease Risk	Target label	Categorical

2.4. Decision Tree CART Model

The classification method used in this study is the Decision Tree using the Classification and Regression Tree (CART) algorithm. CART was used to develop a predictive model based on the patterns of relationships between microclimate parameters and the risk level of chili pepper diseases.

CART uses the Gini index as the criterion for selecting the best split during the decision tree construction process. A lower Gini value indicates a higher degree of data homogeneity.

Gini Index Formula:

$$Gini(t) = \mathbf{1} - \sum_{\mathbf{i=1}}^{\mathbf{k}} (\mathbf{p_i^2})$$

The CART model is built through three main stages: tree growing, pruning, and model validation. The model results are then implemented into an Android system to generate real-time predictions.[9]

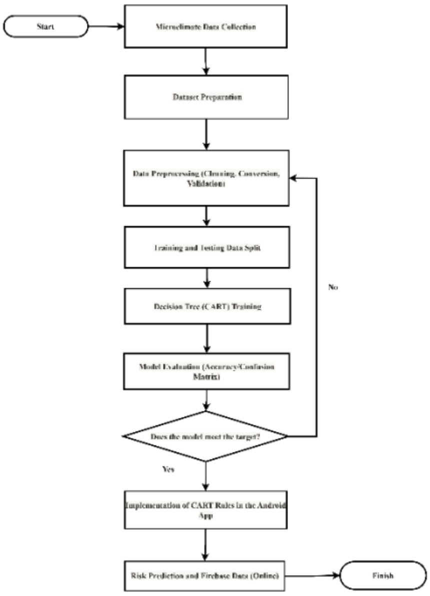


Fig. 4: CART Workflow and Model Training Process

3. Results and Discussion

3.1. System Implementation Results

The implemented system successfully integrates an Internet of Things (IoT) architecture using ESP32, environmental sensors, Firebase Realtime Database, and an Android application. The system is capable of collecting microclimate data from chili plant environments in real-time, including air temperature, air humidity, soil moisture, light intensity, and rainfall conditions.

The ESP32 processes sensor data and transmits it to Firebase via Wi-Fi communication. The Android application retrieves the data from Firebase and displays it in a structured interface for user monitoring. The system also integrates a Decision Tree CART model to classify disease risk levels into low, medium, and high categories.

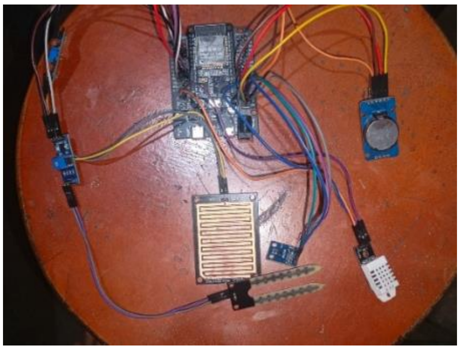


Fig. 5: Implementation of IoT-Based Microclimate Monitoring System

3.2. Sensor Testing Results

The performance of the installed sensors was evaluated to ensure accurate data acquisition for the system. Each sensor was tested under different environmental conditions.

The DHT22 sensor successfully measured air temperature and humidity with stable readings. The capacitive soil moisture sensor provided responsive measurements to changes in soil water content. The BH1750 sensor accurately detected variations in light intensity, while the rain sensor successfully identified wet and dry conditions. The DS3231 RTC module ensured accurate timestamping for each recorded dataset.

These results indicate that all sensors function properly and are suitable for real-time microclimate monitoring in agricultural environments.

Table 2: Sensor Testing Results

No	Sensor	Parameter	Condition	Output Result	System Response
1	DHT22	Air Temperature	Normal room condition	Stable reading (°C)	Data successfully recorded
2	DHT22	Air Humidity	Normal room condition	Stable reading (%RH)	Data successfully recorded
3	Capacitive Soil Moisture	Soil Water Content	Dry soil	Low moisture value	Detected correctly
4	Capacitive Soil Moisture	Soil Water Content	Wet soil	High moisture value	Detected correctly
5	BH1750	Light Intensity	Bright light	High lux value	Detected accurately
6	BH1750	Light Intensity	Low light	Low lux value	Detected accurately
7	Rain Sensor	Rain Condition	Dry condition	No rain detected	System OFF signal
8	Rain Sensor	Rain Condition	Wet condition	Rain detected	System ON signal
9	DS3231 RTC	Time Stamp	Continuous operation	Accurate time logging	Synchronized successfully

3.3. Firebase Realtime Database Performance

Firebase Realtime Database was implemented as a cloud-based storage and synchronization platform between the ESP32 device and the Android application. The database stores control commands, device information, and real-time microclimate monitoring data. This structure allows the system to transmit sensor data from ESP32 and display the latest monitoring results through the Android application.

The Firebase implementation consists of several main nodes, including the control node, device node, and monitoring node. The control node is used to manage system commands, while the device and monitoring nodes store device information and sensor readings. The summary of the database structure and control node is shown in Table 3.

Table 3: Summary of Firebase Database Structure and Control Node

No	Firebase section	Parameter / node	Function	Testing result	Status
1	Main database structure	Root node	Organizes Firebase data into control, device, and monitoring sections	Database structure was displayed properly	Success
2	Control node	resetData	Sends command to reset monitoring data	Value was false, indicating no reset process was running	Success
3	Control node	restartESP32	Sends command to restart the ESP32 device	Value was false, indicating no restart process was running	Success
4	Control node	system	Controls system activation status	Value was true, indicating the monitoring system was active	Success
5	Realtime synchronization	Android–Firebase–ESP32	Enables command exchange between Android application and ESP32	Control data was synchronized through Firebase	Success

Based on Table 3, the Firebase database structure was successfully configured to separate system control and monitoring data. The control node worked properly because each command parameter could represent the system condition clearly. The system parameter showed an active condition, while the reset and restart parameters were inactive during testing. This indicates that Firebase can be used as a real-time communication bridge between the Android application and the ESP32 device.

https://monitoringcabai-a159f-default-rtdb.asia-southeast1.firebasedatabase.app

```
https://monitoringcabai-a159f-default-rtdb.asia-southeast1.firebasedatabase.app/
▸ — Monitoring
▸ — Prediksi
▸ — device
```

```
https://monitoringcabai-a159f-default-rtdb.asia-southeast1.firebasedatabase.app
Prediksi
    riwayat
        -Nv101
            hasil: "Risiko Tinggi"
            timestamp: "2026-05-28 03:05:00"
    terbaru
        hasil: "Risiko Tinggi"
        rekomendasi: "Periksa Drainase"
```

Fig. 6: Firebase Database Structure and Control Node

The next part explains the device node separately. This node is important because it identifies the ESP32 device used in the monitoring system. The device node stores data based on the device identity, so the system can distinguish the data source. In this study, the device identity is represented by esp32_01, which indicates the ESP32 unit used for collecting microclimate data from chili plant environments.

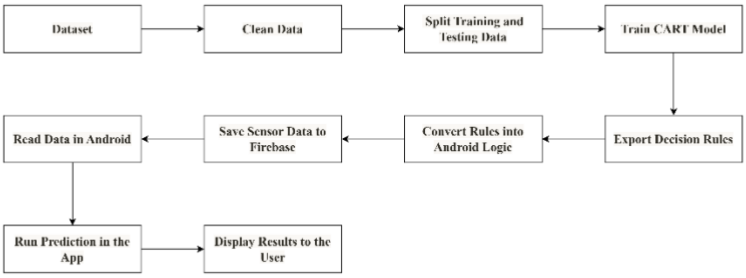


Fig. 7: Device Node in Firebase Realtime Database

The device node shows that the Firebase structure can store data based on the specific IoT device used in the system. This structure supports system scalability because additional ESP32 devices can be added using different device identities. In this implementation, the device node confirms that the monitoring data sent to Firebase comes from the ESP32 device connected to the microclimate sensors.

The monitoring node is used to store real-time sensor readings from the ESP32. This node contains microclimate parameters used by the system, including air temperature, air humidity, soil moisture, light intensity, and rainfall condition. These data are accessed by the Android application to display environmental conditions and support disease risk prediction.

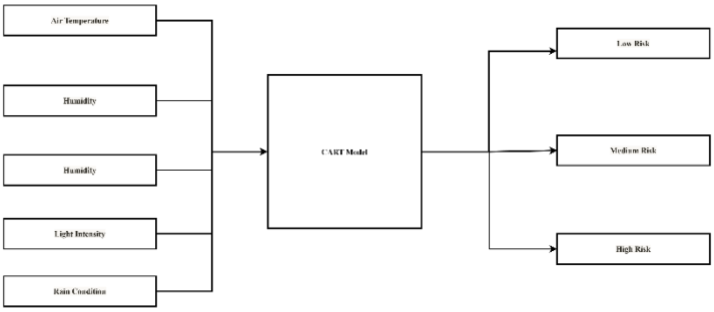


Fig. 8: Monitoring Node in Firebase Realtime Database

The monitoring node shows that sensor data can be stored and updated in Firebase in real time. The stored data represent the latest condition of the chili plant environment. This structure allows the Android application to retrieve current monitoring values and display them to users. The successful update of monitoring data indicates that the communication between ESP32, Firebase, and the Android application works properly.

3.4. Decision Tree CART Model Results

The Decision Tree CART model was implemented to classify chili plant disease risk based on microclimate parameters. The model uses air temperature, air humidity, soil moisture, light intensity, and rainfall condition as input features. The classification output consists of three risk levels: low, medium, and high. The model was trained using labeled dataset and then integrated into the system for real-time prediction.

Table 4. CART Training Dataset

Air Temperature (°C)	Air Humidity (%)	Soil Moisture (%)	Light Intensity (lux)	Rain Conditions	Risk Target
29	70	65	12000	0	Low
30	75	60	11000	0	Low
35	80	55	10000	1	Moderate
32	85	50	9000	1	Moderate
33	90	45	8000	1	High
…..	…..	…..	…..	…..	…..

The results show that the CART model is able to form clear decision rules based on microclimate conditions, making it suitable for implementation in mobile-based agricultural monitoring systems.

3.5. Model Evaluation

The performance of the Decision Tree CART model was evaluated using accuracy, precision, recall, and F1-score metrics. The test results show that the model achieved:

Table 5. Performance Evaluation of CART Mode

No	Evaluation Metric	Value (%)	Description
1	Accuracy	95%	Proportion of correctly classified data from total test data
2	Precision	96%	Ability of the model to correctly predict positive classes
3	Recall	95%	Ability of the model to detect all relevant positive cases
4	F1-Score	95%	Harmonic mean between precision and recall

From 20 test data samples, 19 were correctly classified, indicating high model reliability in predicting chili plant disease risk based on microclimate data.

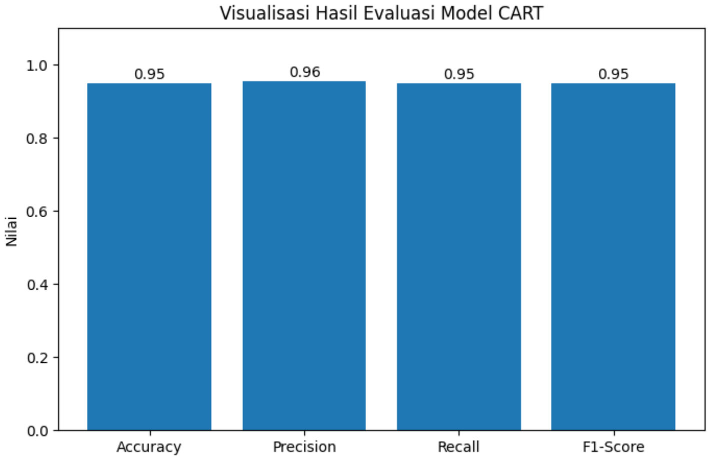


Fig. 9: Evaluation Metrics

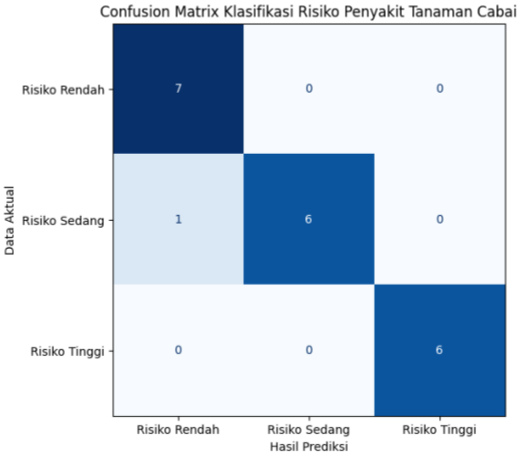


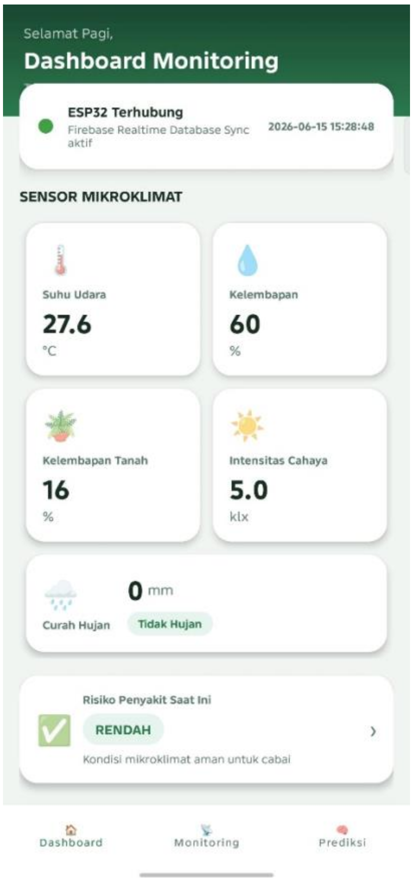
Fig. 10: Confusion Matrix

These results demonstrate that the CART algorithm is effective in handling microclimate-based classification problems and can be reliably integrated into IoT-based agricultural systems.

3.6. Android Application Performance

The Android application was tested to evaluate its functionality in displaying real-time microclimate data and disease risk prediction results. The application successfully retrieved data from Firebase and displayed it in an organized interface. Users can monitor environmental parameters such as temperature, humidity, soil moisture, light intensity, and rainfall status. In addition, the application displays the output of the Decision Tree CART model in the form of disease risk classification.

The results show that the Android application provides a stable and responsive interface for end-user monitoring.



Selamat Pagi,

Dashboard Monitoring

ESP32 Terhubung
Firebase Realtime Database Sync aktif
2026-06-15 15:28:48

SENSOR MIKROKLIMAT

Suhu Udara
27.6
°C

Kelembapan
60
%

Kelembapan Tanah
16
%

Intensitas Cahaya
5.0
klx

0 mm
Curah Hujan
Tidak Hujan

Risiko Penyakit Saat Ini
RENDAH
Kondisi mikroklimat aman untuk cabai

Fig. 11: Android Application Interface

Table 6. Android Application Test Results

No	Tested Feature	Test Scenario	Expected Result	Result
1	Login	The user enters authentication data	The system verifies access and displays the main page	Success
2	Dashboard	The user enters the application	The latest microclimate condition summary is displayed	Success
3	Microclimate Monitoring	The user opens the monitoring page	Air temperature, air humidity, soil moisture, light intensity, and rainfall data are displayed according to Firebase data	Success
4	Monitoring Detail	The user opens the detail page	Historical sensor readings based on time can be displayed	Success
5	Disease Risk Prediction	The application receives the latest microclimate data	The disease risk category, including low, moderate, or high risk, is displayed	Success

Based on Table 4, all main features in the Android application worked according to the system design. The login feature successfully verified user access, while the dashboard and monitoring pages displayed real-time microclimate data from Firebase. The detail page also displayed historical monitoring data based on time records. In addition, the prediction feature successfully presented the disease risk category based on the latest microclimate data received by the application. These results indicate that the Android application can function properly as a user interface for monitoring and disease risk prediction in chili plants.

4. Conclusion

This study developed and implemented an Internet of Things (IoT)-based microclimate monitoring system for chili plants integrated with an Android application and Firebase Realtime Database. The system uses ESP32 as the main controller connected to DHT22, capacitive soil moisture, BH1750, rain sensor, and DS3231 RTC to collect real-time environmental data, including air temperature, air humidity, soil moisture, light intensity, and rainfall conditions. The data are transmitted to Firebase through Wi-Fi and displayed on the Android application, enabling real-time monitoring of plant environmental conditions. The system successfully integrates hardware, cloud services, and mobile applications into a unified monitoring platform that supports agricultural decision-making.

In addition, the Decision Tree Classification and Regression Tree (CART) algorithm was applied to classify chili plant disease risk into low, moderate, and high categories based on microclimate parameters. The model achieved an accuracy of 95%, precision of 96%, recall of 95%, and F1-score of 95%, with 19 out of 20 test data correctly classified. These results indicate that the proposed system is capable of providing accurate and reliable disease risk prediction, making it suitable for supporting precision agriculture and improving decision-making efficiency in chili cultivation.

References

[1] M. Telaumbanua, F. K. Wisnu, S. E. R. Manurung, Y. Erika, B. Lanya, and A. Haryanto, “Enhancing Chili Farming through IoT-Enabled Microclimate Monitoring,” 2024, pp. 175–187. doi: 10.2991/978-94-6463-366-5_17.

[2] M. Telaumbanua *et al.*, “Design of temperature-soil moisture control and monitoring system for chili cultivation in greenhouse,” in *IOP Conference Series: Earth and Environmental Science*, Institute of Physics, 2024. doi: 10.1088/1755-1315/1386/1/012029.

[3] James Erick Lumbantoruan, “Monitoring dan Kontrol Tanaman Cabai berbasis Internet of Things dengan Menggunakan Aplikasi MIT App Inventor,” *Venus: Jurnal Publikasi Rumpun Ilmu Teknik* , vol. 2, no. 6, pp. 179–187, Dec. 2024, doi: 10.61132/venus.v2i6.641.

[4] A. D. Septiadi, Y. I. Sulistya, M. Istighosah, M. Septiara, D. P. Rakhmadani, and A. D. S. Rumestri, “Implementasi IoT Untuk Monitoring Pertumbuhan Tanaman Cabai Dengan Sistem Penyiraman Otomatis Di Desa Kembaran Wetan,” *KREATIF: Jurnal Pengabdian Masyarakat Nusantara*, vol. 5, no. 2, pp. 455–468, Jun. 2025, doi: 10.55606/kreatif.v5i2.6901.

[5] A. Syatriawan, Fadlisyah, and Kurniawati, “PENERAPAN METODE DECISION TREE CART UNTUK KLASIFIKASI PENYAKIT PADA TANAMAN KELAPA SAWIT,” *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab*, vol. 10, no. 2, pp. 1191–1199, Jul. 2025, doi: 10.36341/rabit.v10i2.6544.

[6] G. A. Rohman and A. R. Isnaini, “Otomatisasi Pengendalian Suhu dan Kelembaban Berbasis Internet of Things pada Kandang Ayam Potong,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 5, no. 2, pp. 558–565, Apr. 2025, doi: 10.57152/malcom.v5i2.1686.

[7] F. Puspasari, T. P. Satya, U. Y. Oktiawati, I. Fahrurrozi, and H. Prisyanti, “Analisis Akurasi Sistem Sensor DHT22 berbasis Arduino terhadap Thermohygrometer Standar,” *Jurnal Fisika dan Aplikasinya*, vol. 16, no. 1, p. 33, Feb. 2020, doi: 10.12962/j24604682.v16i1.5717.

[8] A. Sudarmaji *et al.*, “RANCANG BANGUN SISTEM IRIGASI OTOMATIS BERBASIS SENSOR KAPASITIF KELEMBAPAN TANAH,” *Journal of Agricultural and Biosystem Engineering Research*, vol. 5, no. 1, p. 8, May 2024, doi: 10.20884/1.jaber.2024.5.1.11349.

[9] F. Maisa Hana, W. Cholid Wahyudin, S. Ulya, and D. Setia Negara, “IMPLEMENTASI ALGORITMA CART DALAM KLASIFIKASI PENYAKIT DIABETES,” 2023.