

Membangun Desain API Internet of Things pada Usaha Tani Kentang

Mai Fernando Nainggolan*¹, Surya Abadi Sembiring²

^{1,2} Program Studi Agribisnis, Fakultas Pertanian, Unika Santo Thomas

E-mail: mai_fernando@ust.ac.id

ABSTRAK

Internet Of Things adalah suatu teknologi yang memungkinkan benda-benda terhubung benda lainnya atau user melalui internet. Di bidang pertanian, internet of things memungkinkan petani dapat memantau kondisi nutrisi pada tanamannya dan kondisi lapangan. Tujuan dari penelitian ini adalah untuk memudahkan petani dalam melakukan pemupukan dan penyiraman tanaman, mencatat perkembangan tanaman, dan mengecek kondisi nutrisi pada Usaha Tani dan kondisi fisiologis di lapangan secara *real-time*. Petani mendapatkan informasi tanah dan fisiologis lapangan dari sensor tanah dan sensor lapangan sudah dipasang pada tanah. Data tersebut dikirim oleh sensor ke database MySQL melalui API yang diakses oleh mikrokontroler ESP32 dan dapat ditampilkan pada aplikasi mobile. Dari beberapa eksperimen yang dilakukan, hasil menunjukkan bahwa seluruh data yang diakses oleh mikrokontroler menghasilkan 100% berupa bentuk JSON.

Kata Kunci: Internet of Things, Petani, Nutrisi Tanaman, Mikrokontroler, API, JSON.

ABSTRACT

The Internet of Things is a technology that allows objects to connect to other objects or users via the Internet. In the agricultural sector, the Internet of Things allows farmers to monitor the nutritional conditions of their plants and field conditions. The purpose of this research is to make it easier for farmers to fertilize and water plants, record plant development, and check nutritional conditions in agricultural businesses and physiological conditions in the field in real-time. Farmers obtain soil and field physiological information from soil sensors and field sensors that have been installed in the soil. The data is sent by the sensor to a MySQL database via an API accessed by the ESP32 microcontroller and can be displayed on a mobile application. From several experiments conducted, the results show that all data accessed by the microcontroller produces 100% in JSON format.

Keywords: Keywords: Internet of Things, Farmer, Plant Nutrition, Microcontroller, API, JSON.

PENDAHULUAN

Kabupaten Simalungun merupakan salah satu wilayah potensial di Provinsi Sumatera Utara yang memiliki kondisi agroklimat mendukung bagi budidaya kentang (Martauli, 2021). Tanaman kentang menjadi komoditas penting bagi petani di dataran tinggi Simalungun, terutama karena nilai ekonominya yang tinggi serta permintaan pasar yang terus meningkat (Siregar, 2024). Meskipun demikian, praktik budidaya kentang di wilayah ini masih menghadapi berbagai tantangan, terutama terkait efisiensi produksi, ketersediaan data lapangan secara real-time, dan kendala dalam pengambilan keputusan budidaya yang tepat waktu (Pangestika et al, 2020).

Seiring dengan perkembangan teknologi digital, pendekatan berbasis Internet of Things (IoT) mulai diperkenalkan dalam sektor pertanian untuk menjawab permasalahan tersebut. IoT memungkinkan integrasi antara sensor lingkungan, perangkat pengontrol, dan sistem pengolahan data secara otomatis (Wardhana, 2025). Dalam konteks usaha tani kentang, IoT dapat memberikan informasi akurat terkait suhu tanah, kelembaban udara, tingkat curah hujan, hingga kebutuhan irigasi dan pemupukan. Namun, pemanfaatan IoT di Kabupaten Simalungun masih tergolong rendah dan belum terdesain secara sistematis sesuai kebutuhan lokal.

Salah satu kunci keberhasilan dalam penerapan sistem IoT di bidang pertanian adalah adanya Application Programming Interface (API) yang efektif (Maharani, 2021).

API berfungsi sebagai jembatan antara perangkat keras IoT (sensor, aktuator) dengan perangkat lunak pengelola data (dashboard, aplikasi mobile, dan sistem pengambilan keputusan). Dengan API yang terstruktur dan terbuka, proses pertukaran data menjadi lebih efisien, terstandarisasi, dan mudah dikembangkan lebih lanjut (Karna, 2021). Sayangnya, belum banyak studi atau desain sistem API yang dirancang khusus untuk pertanian kentang di kawasan seperti Simalungun.

Pengembangan desain API IoT pada usaha tani kentang tidak hanya berorientasi pada teknologi, tetapi juga harus mempertimbangkan aspek sosial, ekonomi, dan kapasitas digital petani lokal. Sebagian besar petani di Kabupaten Simalungun masih beradaptasi dengan penggunaan perangkat digital, sehingga desain sistem harus bersifat adaptif, mudah digunakan, dan memberikan manfaat langsung terhadap produktivitas serta efisiensi biaya produksi. Ini menuntut adanya pendekatan partisipatif dan pemetaan kebutuhan lapangan secara komprehensif.

Selain itu, integrasi data melalui API dapat membuka peluang besar bagi pengembangan sistem pertanian presisi (precision agriculture) berbasis data historis dan prediktif. Data-data lapangan yang dikumpulkan melalui sensor dapat diolah menggunakan machine learning untuk memprediksi waktu tanam optimal, risiko serangan hama, maupun prediksi hasil panen. Semua proses ini sangat bergantung pada arsitektur API yang mampu menjamin

kecepatan, akurasi, dan keamanan pertukaran data antar sistem.

Dalam konteks kebijakan daerah, upaya transformasi digital pertanian seperti ini juga sejalan dengan arah pembangunan pertanian modern yang dicanangkan pemerintah. Kabupaten Simalungun memiliki peluang besar untuk menjadi model penerapan smart farming berbasis teknologi lokal dengan tetap mengedepankan kearifan lokal. Desain API yang sesuai dengan karakteristik wilayah dan kebutuhan petani akan memperkuat sistem pertanian berkelanjutan di masa depan (Sari, 2024).

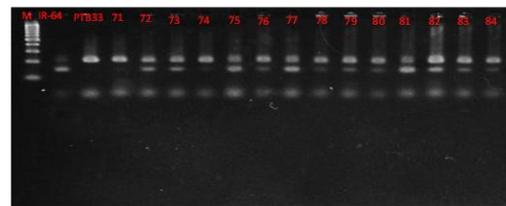
Oleh karena itu, membangun desain API Internet of Things pada usaha tani kentang di Kabupaten Simalungun menjadi langkah strategis dalam mendorong efisiensi dan modernisasi sektor pertanian. Penelitian dan pengembangan yang diarahkan pada integrasi antara teknologi dan praktik budidaya lokal akan menghasilkan solusi yang aplikatif dan berdampak langsung terhadap peningkatan kesejahteraan petani.

Dengan memahami pentingnya desain sistem yang kontekstual, terstandarisasi, dan berkelanjutan, artikel ini akan membahas secara mendalam perancangan API untuk mendukung sistem IoT pada usaha tani kentang di Simalungun, termasuk tantangan teknis dan sosialnya, serta potensi pengembangan ke depan.

Metode Penelitian

2.1 Arsitektur Rest API IoT pada Usaha Tani Kentang

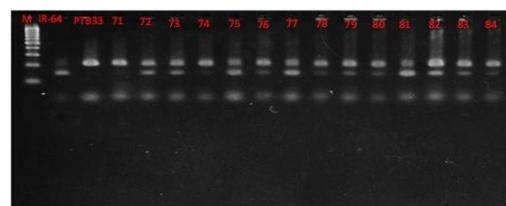
Arsitektur REST API IoT merupakan kerangka komunikasi antara perangkat IoT dengan aplikasi melalui protokol HTTP. Perangkat IoT dapat berupa sensor dan aktuator yang akan meneruskan permintaan dari aplikasi yang direpresentasikan dengan method GET dan POST. Aplikasi mengirim permintaan dan menerima hasil permintaan dari database dalam format data JavaScript Object Notation (JSON). Arsitektur REST API IoT ditunjukkan pada Gambar 1 di bawah ini.



Gambar 1. Arsitektur REST API IoT pada Usaha Tani Kentang.

2.2 Arsitektur Sistem IoT pada Usaha Tani Kentang

Arsitektur Sistem IoT adalah serangkaian alur yang merepresentasikan pengembangan perangkat IoT menjadi satu kesatuan sistem. Pada penelitian ini, desain arsitektur sistem pada Usaha Tani Kentang ditunjukkan pada Gambar 2 di bawah ini.



Gambar 2. Arsitektur Sistem IoT pada Usaha Tani Kentang.

Pada Gambar 2, arsitektur sistem IoT dibagi menjadi 4 bagian, yaitu perangkat IoT yaitu sensor yang dipasang di lahan Kentang, perangkat IoT berupa aktuator yang melakukan perintah pemupukan dan penyiraman, dan perangkat IoT yang memproses permintaan aplikasi melalui protokol HTTP, serta perangkat yang menampilkan data sensor secara realtime dan melakukan pemupukan dan penyiraman berupa aplikasi berbasis mobile. Alur arsitektur sistem dimulai data nutrisi tanah yang disimpan di database secara real time. Aplikasi berbasis mobile menampilkan data nutrisi datanah menggunakan API melalui protokol HTTP. Berdasarkan data nutrisi tanah, user melakukan perintah pemupukan dan penyiraman dengan mengirim perintah siram menggunakan API penyiraman. Proses ini dimulai dengan memasukkan data jumlah nutrisi nitrogen, fosfor, kalium dan air dan kemudian mengirimkan data melalui API dalam format JSON dengan metode POST. Data penyiraman diproses, diteruskan ke aktuator dan disimpan di database. Aktuator melakukan konversi jumlah liter penyiraman ke dalam satuan waktu dan melakukan penyiraman berdasarkan waktu yang tersedia. Jika proses pemupukan dan penyiraman selesai, maka aktuator akan mati secara otomatis. Selain itu, user juga melakukan pencatatan perkembangan Usaha Tani Kentang melalui API perkembangan yang telah disediakan.

2.3 Use Case Diagram

Use case diagram adalah pemodelan sistem yang menggambarkan kemampuan sistem itu sendiri dalam memfasilitasi

kebutuhan user. Pada penelitian ini, user dapat membaca data dari sensor secara real time, dan juga melakukan perintah pemupukan dan penyiraman pada Usaha Tani Kentang. User juga dapat mencatat perkembangan Usaha Tani setiap saat. Use case diagram untuk desain API IoT pada Usaha Tani Kentang ditunjukkan pada Gambar 3.

2.4 Entity Relationship Diagram

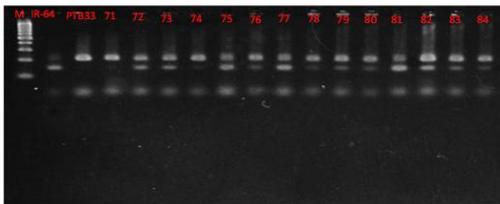
Entity Relationship Diagram adalah diagram yang merepresentasikan hubungan antar atribut data yang saling berelasi melalui tabel data tersebut. Pada penelitian ini, Desain API yang dibangun pada sistem IoT terdiri dari 6 tabel, yaitu: tabel user, larik pohon, sensor lapangan, sensor realtime, penyiraman, dan perkembangan. ERD ditunjukkan pada Gambar 4 di bawah ini.

Pada Gambar 4, tabel user memiliki relasi dengan tabel larik pohon dengan kardinalitas 1:n dimana 1 user dapat menginput data lebih dari 1 larik pohon. Tabel user memiliki relasi dengan tabel penyiraman dengan kardinalitas 1:n dimana 1 user dapat melakukan penyiraman lebih dari 1 kali. Tabel user juga memiliki relasi dengan tabel perkembangan dengan kardinalitas 1:n dimana 1 user dapat mencatat data perkembangan pohon lebih dari 1 kali. Tabel larik pohon memiliki relasi dengan sensor real time dengan kardinalitas 1:n dimana 1 larik pohon dapat memiliki lebih dari 1 data sensor real time. Tabel larik pohon juga memiliki relasi dengan tabel penyiraman dengan kardinalitas 1:n

Dimana 1 larik pohon dapat memiliki lebih dari 1 data penyiraman.

2.5 Class Diagram

Class diagram merupakan diagram yang menggambarkan diagram struktur statis yang mendeskripsikan atribut, metode dan relasi antar objek. Class diagram yang diusulkan pada penelitian ini ditunjukkan pada Gambar 5.



Gambar 5. Class Diagram yang Diusulkan.

Pada Gambar 5 menjelaskan setiap entitas dan fungsinya. Entitas ini merepresentasikan tabel yang memiliki atribut dan metodenya. Setiap entitas memiliki alur pengiriman datanya melalui proses yang terjadi pada Pertanian Cerdas. Proses yang terjadi direpresentasikan dalam bentuk flowchart diagram. Pada penelitian ini terdapat 5 flowchart yang merepresentasikan permintaan dari aplikasi Mobile melalui API yang dibuat oleh backend system. Flowchart permintaan data detail nutrisi tanah pada Usaha Tani ditunjukkan pada Gambar 6.

Pada Gambar 6, permintaan data detail nutrisi tanah pada Usaha Tani dimulai dari Aplikasi Mobile mengirim permintaan dengan metode GET melalui API. Kemudian Backend system akan mengecek permintaan. Jika data ada, maka data detail pohon dan nutrisi tanahnya akan ditampilkan secara real time

dalam bentuk API dan ditampilkan pada Aplikasi Mobile, namun jika tidak ada, maka akan menampilkan pesan data kosong. Sedangkan flowchart untuk proses melakukan pemupukan dan penyiraman ditunjukkan pada Gambar 7.

Pada Gambar 7 mendeskripsikan permintaan dari Aplikasi Mobile untuk melakukan perintah pemupukan dan penyiraman dengan metode POST melalui API. Dengan asumsi User telah memilih salah satu larik pohon, kemudian pada Aplikasi Mobile, user harus mengisi data masing-masing nutrisi nitrogen, fosfor, kalium dan air, kemudian mengirim permintaan dengan metode POST melalui API. Sistem backend akan mengeksekusi permintaan dengan cara mengkonversi jumlah liter penyiraman kedalam waktu, lalu menggerakkan aktuatur ooooo untuk melakukan pemupukan dan penyiraman. Data pemupukan dan penyiraman akan disimpan ke database. User dapat menampilkan data pemupukan dan penyiraman berdasarkan larik pohon pada Aplikasi Mobile dengan metode GET melalui API. Flowchart mencatat perkembangan Usaha Tani ditunjukkan pada Gambar 8.

Gambar 8 mendeskripsikan bagaimana proses permintaan untuk mencatat perkembangan Usaha Tani Kentang pada Aplikasi Mobile. Dengan asumsi pada aplikasi mobile user membuka menu Perkembangan, menekan tombol Tambah Data, user harus mengisi diameter batang dan tinggi pohon, lalu mengirim permintaan dengan metode POST

melalui API. Sistem backend akan memproses permintaan pencatatan dengan menghitung biomassa dan Produksi minimum dan maksimum menggunakan rumus yang sudah disisipkan pada API. Kemudian pada Aplikasi Mobile user dapat menampilkan data perkembangan Usaha Tani masing-masing pohon yang sudah dicatat, dengan memasukkan nomor pohon menggunakan method POST melalui API.

HASIL DAN PEMBAHASAN

3.1 Implementasi Kode Program

Kode program adalah implementasi dari rancangan class diagram berdasarkan flowchart yang ada. Setiap kode program yang diimplementasikan dibungkus dalam wadah fungsi. Setiap fungsi akan menjalankan perintah berdasarkan permintaan dari Aplikasi Mobile dan menampilkannya dalam bentuk API dengan format data JSON dalam method POST maupun GET. Kode program untuk permintaan data detail nutrisi tanah ditunjukkan pada Gambar 9.

```
public function getByPohon($pohon_id)
{
    $sensors = SensorRealtime::where('pohon_id', $pohon_id)
    ->orderByDesc('created_at')
    ->orderByDesc('id')->get();
    return response()->json($sensors);
}
```

Gambar 9. Kode program permintaan data detail nutrisi tanah.

Pada Gambar 9, kode program dibungkus dalam function getByPohon yang mengeksekusi permintaan data detail nutrisi tanah berdasarkan larik pohon. Keluaran dari fungsi ini adalah data detail nutrisi tanah pada

tanaman, sehingga pada aplikasi Mobile, hanya menampilkan index ke-0 yang merupakan data pertama. Kode program melakukan pemupukan dan penyiraman adalah kode program untuk mengeksekusi perintah pemupukan dan penyiraman pada larik pohon yang dipilih. Kode program pemupukan dan penyiraman ditunjukkan pada Gambar 10.

```
public function turnOn(Request $request)
{
    $nitrogen = $request->nitrogen;
    $fosfor = $request->fosfor;
    $kalium = $request->kalium;
    $air = $request->air;
    $total_air = $nitrogen + $fosfor + $kalium + $air;
    $total_waktu = round($total_air / 100,0);

    $penyiraman = new Penyiraman();
    $penyiraman->pohon_id = $request->pohon_id;
    $penyiraman->nitrogen = $nitrogen;
    $penyiraman->fosfor = $fosfor;
    $penyiraman->kalium = $kalium;
    $penyiraman->air = $air;
    $penyiraman->durasi_waktu = $total_waktu;
    $penyiraman->is_on = 1;
    $penyiraman->save();

    sleep($total_waktu);

    $penyiraman_update = Penyiraman::find($penyiraman->id);
    $penyiraman_update->is_on = 0;
    $penyiraman_update->save();

    return response()->json([
        'success' => true,
        'message' => 'Penyiraman Selesai',
        'data' => $penyiraman_update
    ], 201);
}
```

Gambar 10. Kode program pemupukan dan penyiraman tanaman.

Pada Gambar 10, kode program pemupukan dan penyiraman dibungkus dalam function turnOn() yang mengeksekusi proses pemupukan dan penyiraman. Terdapat beberapa atribut yang wajib diisi, yaitu larik pohon, jumlah nitrogen, jumlah fosfor, jumlah kalium, jumlah air. Jumlah nitrogen, fosfor, kalium dan air akan dihitung secara dijumlahkan. Lalu totalnya akan dikonversi menjadi satuan waktu. Output dari kode program ini adalah mengembalikan data status

pemupukan dan penyiraman menjadi Off dan data pemupukan dan penyiraman. dalam bentuk API.

Kode program mencatat perkembangan Usaha Tani adalah kode program yang mencatat perkembangan Usaha Tani setelah user mengisi beberapa atribut seperti nomor pohon, diameter batang (dalam satuan cm) dan tinggi pohon (dalam satuan m). Kode program mencatat perkembangan Usaha Tani ditunjukkan pada Gambar 11.

```
public function store(Request $request)
{
    $biomassa = 0;
    $volume_atsiri_min = 0;
    $volume_atsiri_max = 0;
    $skadar_atsiri_min = 1/100;
    $skadar_atsiri_max = 2/100;
    $validator = Validator::make($request->all(), [
        'tanggal_periksa' => 'required',
        'pohon_id' => 'required',
        'diameter' => 'required',
        'tinggi' => 'required',
    ]);
    $diameter = $request->diameter;
    $tinggi = $request->tinggi;
    $a = 0.8597;
    $b = 2.622;
    $c = 0.515;
    $biomassa = $a * pow($diameter, $b) * pow($tinggi, $c);
    $biomassa = round($biomassa, 2);
    $volume_atsiri_min = $biomassa * $skadar_atsiri_min;
    $volume_atsiri_max = round($volume_atsiri_min, 2);

    $volume_atsiri_max = $biomassa * $skadar_atsiri_max;
    $volume_atsiri_max = round($volume_atsiri_max, 2);

    if ($validator->fails()) {
        return response()->json($validator->errors(), 422);
    }
    $model = new Perkembangan();
    $model->tanggal_periksa = $request->tanggal_periksa;
    $model->pohon_id = $request->pohon_id;
    $model->diameter = $diameter;
    $model->tinggi = $tinggi;
    $model->biomassa = $biomassa;
    $model->volume_atsiri_min = $volume_atsiri_min;
    $model->volume_atsiri_max = $volume_atsiri_max;
    $perkembangan = [
        'tanggal_periksa' => $request->tanggal_periksa,
        'pohon_id' => $request->pohon_id,
        'diameter' => $diameter,
        'tinggi' => $tinggi,
        'biomassa' => $biomassa,
        'volume_atsiri_min' => $volume_atsiri_min,
        'volume_atsiri_max' => $volume_atsiri_max
    ];
    $model->save();
    return new PerkembanganResource(true, 'Data Perkembangan Berhasil Ditambahkan!', $perkembangan);
}
```

Gambar 11. Kode program mencatat perkembangan tanaman.

Pada Gambar 11, kode program mencatat perkembangan Usaha Tani dibungkus dalam function store (). Kode program ini mengeksekusi inputan user dengan menghitung jumlah biomassa menggunakan rumus alometrik. Jumlah biomassa yang didapatkan digunakan untuk menghitung prediksi Jumlah

Produksi minimum dan maksimum menggunakan rumus yang sudah ditentukan.

3.2 Keluaran Kode Program

Setiap hasil permintaan data dari Aplikasi Mobile akan dikembalikan dan ditampilkan oleh Backend System dalam format API JSON. Setiap API yang disediakan oleh Backend System memiliki alamat url endpoint. Setiap endpoint memiliki method seperti: GET, POST, PUT, dan DELETE. Masing-masing method endpoint ini sudah didesain berdasarkan fungsinya pada kode program. Setiap endpoint method GET diujicoba menggunakan web browser. Setiap endpoint method POST diujicoba menggunakan aplikasi Postman.

Permintaan Data Detail Nutrisi pada Tanaman

Pada permintaan data detail nutrisi tanaman, backend system menyediakan endpoint untuk menampilkan data nutrisi berdasarkan larik pohon yang dipilih. Endpoint yang disediakan adalah:

*domain.com/api/sensor-
realtime/pohon/{larik_pohon}*

Endpoint ini diakses dengan method GET, kemudian menampilkan data dalam bentuk API JSON. Setiap larik pohon yang diakses memiliki data id pohon yang berbeda-beda. Atribut id pohon yang berbeda-beda merepresentasikan nutrisi setiap pohon yang berbeda pada setiap larik yang dipilih. Setiap pohon memiliki banyak data nutrisi karena dihasilkan oleh sensor secara real time. User hanya perlu mengambil data nutrisi dengan

index ke-0, yang merupakan informasi nutrisi terbaru dari sensor. Output endpoint ini ditunjukkan pada Gambar 12.



Gambar 12. Permintaan data detail nutrisi pada tanaman.

Permintaan melakukan Pemupukan dan Penyiraman

Pada permintaan melakukan pemupukan dan penyiraman, backend system menyediakan endpoint untuk memproses permintaan setelah user mengisi beberapa atribut parameter, yaitu: larik pohon, jumlah nitrogen, jumlah fosfor, jumlah kalium, dan jumlah air. Endpoint yang disediakan adalah:

domain.com/api/penyiraman/on

Endpoint ini diakses dengan method POST dan akan menghasilkan output dalam bentuk API JSON. Output endpoint ini ditunjukkan pada Gambar 13.



Gambar 13. Permintaan melakukan pemupukan dan penyiraman.

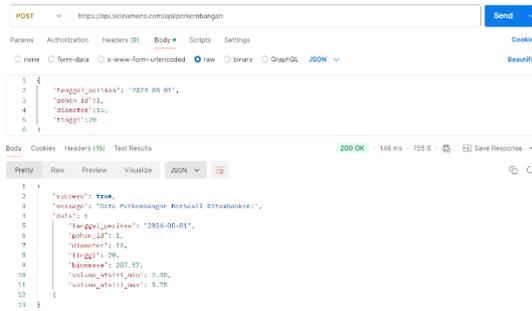
Pada Gambar 13, poin a menjelaskan input parameter pemupukan dan penyiraman menggunakan format JSON. Sebagai contoh, user mengisi parameter pohon id, nitrogen, fosfor, kalium dan air, lalu mengirim permintaan. Backend system memproses permintaan dengan menjumlahkan total nitrogen, fosfor, kalium dan air lalu mengkonversi ke dalam satuan waktu detik. Selama waktu pemupukan dan penyiraman berlangsung, System Backend masih mengirim request seperti pada poin a. Point b menjelaskan output yang dihasilkan setelah waktu proses pemupukan dan penyiraman selesai.

Permintaan mencatat Perkembangan Tanaman

Pada permintaan mencatat perkembangan tanaman, backend system menyediakan endpoint untuk memproses permintaan setelah user mengisi beberapa atribut parameter, yaitu: tanggal periksa, nomor pohon, diameter batang dan tinggi pohon. Endpoint yang disediakan adalah:

domain.com/api/perkembangan

Endpoint ini diakses dengan method POST dan akan menghasilkan output dalam bentuk API JSON. Output endpoint ini ditunjukkan pada Gambar 14.



Gambar 14. Permintaan mencatat perkembangan tanaman.

Pada Gambar 14, input parameter perkembangan Usaha Tani menggunakan format JSON. Sebagai contoh, user mengisi parameter tanggal periksa, pohon id, diameter dan tinggi, lalu mengirim permintaan. Backend system memproses permintaan dengan menghitung jumlah biomassa menggunakan rumus alometrik. Setelah mendapatkan jumlah biomassa, selanjutnya menghitung prediksi Jumlah Produksi minimum dan maksimum menggunakan rumus yang sudah ditentukan. Output dari endpoint ini berupa status tambah data perkembangan berhasil dan data perkembangan yang sudah diinput

KESIMPULAN

Berdasarkan analisis sistem, pengembangan kode program dan pengujian menggunakan Postman, dihasilkan sekumpulan API yang lengkap dan memenuhi kebutuhan user dalam menyelesaikan permasalahan terkait pemantauan kondisi nutrisi tanaman, melakukan pemupukan dan penyiraman, serta pencatatan perkembangan tanaman. Desain API dari Backend system yang sudah dibuat telah diujicoba menggunakan method GET maupun POST. Proses mengakses endpoint dan pengisian parameter yang dibutuhkan pada

setiap endpoint telah menghasilkan output JSON yang diharapkan. Pada pengembangan selanjutnya adalah penyempurnaan sistem dengan mengembangkan Aplikasi berbasis mobile yang menyesuaikan kebutuhan pertanian lebih lanjut, seperti pertanian presisi, penerapan deep learning, machine learning dan artificial intelligent pada tanaman.

DAFTAR PUSTAKA

- Karna, N. B. A., & Mayasari, R. (2021). Perancangan Dan Implementasi IoT Platform Untuk Pemantauan Tanaman Pakcoy Pada Tahap Penyemaian. *eProceedings of Engineering*, 8(2).
- Martauli, E. D. (2021). Analisis Komoditas Unggulan Sektor Pertanian Dataran Tinggi Sumatera Utara. *Agrifor: Jurnal Ilmu Pertanian dan Kehutanan*, 20(1), 123-138.
- Maharani, M. D. D., & Pamungkas, R. W. P. (2021). Internet of Things dan Keberlanjutan Pendekatan Revolusioner Sebaran Petugas Lapangan Pertanian sebagai Enumerator. *Teknotan: Jurnal Industri Teknologi Pertanian*, 15(1), 59-68.
- Pangestika, M., Hohary, M., Agus, Y. H., Widyawati, N., Herawati, M. M., Sutrisno, A. J., ... & Nuswantara, B. (2020). *Smart Farming: Pertanian di Era Revolusi Industri 4.0*. Penerbit Andi.
- Sari, F. P., Munizu, M., Rusliyadi, M., Nuryanneti, I., & Judijanto, L. (2024). *Agribisnis: Strategi, inovasi dan*



keberlanjutan. PT. Green Pustaka
Indonesia.

Siregar, A. (2024). Analisis Pendapatan,
Kelayakan Dan Saluran Pemasaran
Usahatani Hortikultura Bawang Merah
Dan Kentang.

Wardhana, A. S., Ferdiansyah, M., & Kholifah,
S. (2025). Desain dan Prototipe Integrasi
IoT dalam Pertanian Hidroponik Cerdas
Berbasis Energi Terbarukan. Jurnal
Indonesia: Manajemen Informatika Dan
Komunikasi, 6(1), 105-114.