

Rancang Bangun Back-End Sistem Informasi Pertanian Berbasis Website di Kabupaten Sumbawa *(Back-End Development of a Web-Based Agricultural Information System in Sumbawa)*

As'ad Muzakki Nurjamil¹

¹*Teknik Informatika, Universitas Mataram Jln. Majapahit 62, Mataram, Indonesia*

Corresponding author: f1d021028@student.unram.ac.id

(Submitted: May 2025 / Reviewed: June 2025 / Accepted: June 2025)

ABSTRACT

The digitalization of the agricultural reporting and complaint system in Sumbawa Regency through the development of the Sumbawa Agricultural Governance Information System (SiTAWA) addresses longstanding inefficiencies found in manual processes. SiTAWA introduces innovations through the direct integration of farmers, sub-district officials, and the Department of Agriculture within a single, responsive digital platform. The system features a web-based backend designed to build structured APIs and a user-friendly frontend to facilitate seamless interaction among stakeholders. Its added value lies in real-time monitoring of harvest reports and complaint submissions, automated notifications, and an analytical dashboard that supports data-driven decision-making. Compared to previous manual or partially digital systems, SiTAWA offers a standardized and auditable workflow. All functionalities have been tested using black box testing methods, showing results aligned with user expectations. The implementation of this system has proven to improve work efficiency, information transparency, and service quality within the Sumbawa Department of Agriculture.

Keywords:

*Agriculture;
API;
Back-end;
Digitalization;
Information System;*

ABSTRAK

Digitalisasi sistem pelaporan dan pengaduan pertanian di Kabupaten Sumbawa melalui pengembangan Sistem Informasi Tata Kelola Pertanian Sumbawa (SiTAWA) menjadi solusi atas kendala efisiensi, akurasi, dan transparansi yang selama ini dihadapi dalam proses manual. SiTAWA menawarkan inovasi berupa integrasi langsung antara petani, perangkat kecamatan, dan Dinas Pertanian melalui satu platform digital yang responsif. Sistem ini dilengkapi dengan *back-end* berbasis web untuk membangun API secara terstruktur serta *front-end* yang ramah pengguna guna memudahkan interaksi semua pihak. Nilai tambah dari SiTAWA terletak pada fitur pelaporan hasil panen dan pengajuan pengaduan yang dapat dipantau secara real-time, sistem notifikasi otomatis, serta dashboard analitik untuk mendukung pengambilan keputusan berbasis data. Dibanding sistem manual maupun digital sebelumnya yang bersifat parsial dan tidak terintegrasi, SiTAWA menghadirkan alur kerja yang terstandarisasi dan dapat diaudit. Seluruh fungsi diuji menggunakan metode black box testing dan menunjukkan hasil yang sesuai dengan ekspektasi pengguna. Implementasi sistem ini terbukti meningkatkan efektivitas kerja, transparansi informasi, dan kualitas pelayanan di Dinas Pertanian Sumbawa.

Kata kunci:

*Agrikultur;
API;
Back-end;
Digitalisasi;
Sistem Informasi;*

1. Pendahuluan

Dalam era transformasi digital, pemanfaatan teknologi informasi dan komunikasi (TIK) telah menjadi strategi utama dalam meningkatkan efisiensi dan transparansi pelayanan publik, termasuk di sektor pertanian yang menjadi tulang punggung perekonomian masyarakat Kabupaten Sumbawa. Dinas Komunikasi dan Informatika (Kominfo) bersama Dinas Pertanian Kabupaten Sumbawa menyadari pentingnya modernisasi sistem kerja

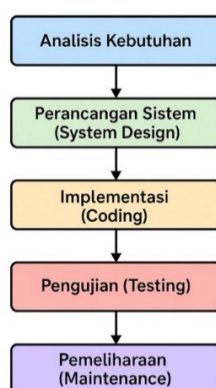
melalui adopsi solusi digital, mengingat masih banyak proses pelayanan, seperti pengaduan masyarakat dan pelaporan hasil pertanian, yang dilakukan secara manual dan tidak efisien.

Website sebagai salah satu bentuk sistem informasi berbasis web memainkan peran penting dalam *e-government* karena mampu menyediakan layanan publik yang lebih terbuka, responsif, dan terintegrasi (Neves & Silva, 2021). Sistem Informasi Pertanian Sumbawa (SiTAWA) dirancang sebagai platform digital yang mencakup kanal pengaduan online, sistem pelaporan pertanian bulanan, serta dashboard analitik berbasis data real-time untuk mendukung pengambilan keputusan. Pengembangan dan pengujian sistem ini juga memerlukan pendekatan metodologis yang tepat, salah satunya melalui pengujian *black-box* yang berfokus pada evaluasi fungsi sistem berdasarkan input dan output tanpa memeriksa struktur internal kode, sehingga efektif dalam mendeteksi kesalahan fungsional dan antarmuka (Khoirudda'I Hermawan et al., 2022; Wijaya & Astuti, 2021)

Studi global dan nasional menunjukkan bahwa digitalisasi sektor pertanian mampu meningkatkan kualitas pelayanan pemerintah daerah dalam pengelolaan data, pelaporan, dan komunikasi antara petani dan lembaga terkait (Bournaris, 2020; Mushi et al., 2022). Oleh karena itu, pengembangan SiTAWA diharapkan menjadi solusi strategis untuk mendukung tata kelola pertanian yang lebih efisien, transparan, dan berbasis teknologi di Kabupaten Sumbawa.

2. Metodologi

Penelitian ini menggunakan model pengembangan perangkat lunak **Waterfall**, yang merupakan pendekatan sistematis dan terstruktur dalam membangun sistem informasi. Model ini dipilih karena cocok digunakan dalam proyek yang memiliki kebutuhan yang sudah jelas dan terdefinisi sejak awal. Tahapan dalam model *Waterfall* mencakup lima langkah utama (Ahmad et al., 2019; Asrin & Utami, 2024), seperti terlihat pada gambar 1 dibawah ini:



Gambar 1. Alur metode *Waterfall*

Penelitian ini menggunakan metode *Waterfall* sebagai model pengembangan perangkat lunak, dengan tahapan yang terstruktur dan berurutan. Gambar 1 menunjukkan diagram alur lengkap dari model *Waterfall*, yang terdiri atas lima tahap utama:

1. Analisis Kebutuhan: Mengidentifikasi permasalahan pada proses pelaporan dan pengaduan manual di Dinas Pertanian Kabupaten Sumbawa serta merumuskan kebutuhan sistem yang akan dikembangkan.
2. Perancangan Sistem (*Design*): Membuat desain antarmuka pengguna (*user interface*) dan struktur basis data untuk mendukung pengelolaan data pengaduan, laporan hasil panen, serta rekapitulasi laporan oleh dinas.
3. Implementasi (*Coding*): Mengembangkan sistem berbasis web menggunakan bahasa pemrograman PHP dan framework Laravel, termasuk integrasi dengan Supabase untuk pengelolaan data.

4. Pengujian (*Testing*): Menggunakan metode *black-box testing* untuk menguji setiap fungsi sistem berdasarkan input dan output, tanpa melihat kode sumber secara langsung, guna memastikan sistem bekerja sesuai kebutuhan pengguna.
5. Pemeliharaan (*Maintenance*): Melakukan penyempurnaan dan perbaikan sistem berdasarkan hasil uji coba dan umpan balik dari pengguna lapangan (Dinas dan perangkat kecamatan).

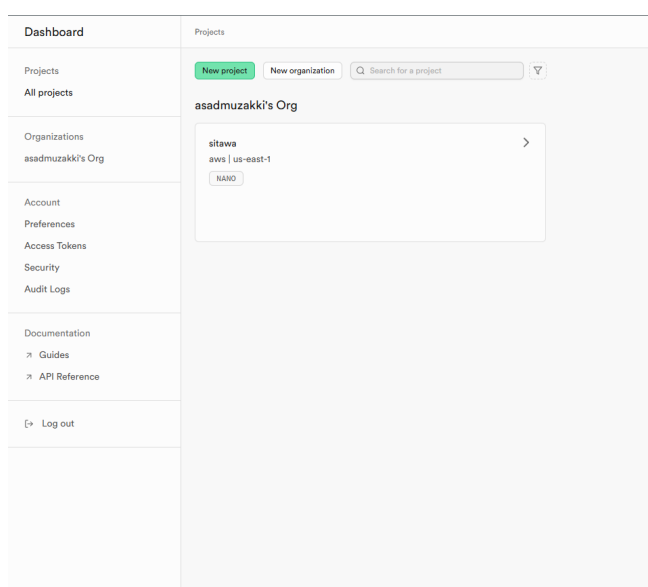
Kelebihan model *Waterfall* dalam konteks penelitian ini adalah memberikan struktur yang jelas pada tiap tahap pengembangan, sehingga cocok untuk kebutuhan proyek yang ruang lingkupnya telah ditentukan sejak awal. Pendekatan ini juga memudahkan koordinasi antara tim pengembang dengan pihak Dinas, karena setiap fase dapat didokumentasikan dan dievaluasi secara sistematis sebelum berlanjut ke tahap berikutnya. Dengan adanya batasan waktu dan sumber daya, *Waterfall* menjadi pilihan tepat untuk menjamin ketuntasan dan kestabilan sistem sebelum diimplementasikan di lingkungan kerja sebenarnya.

3. Hasil dan Pembahasan

3.1. Implementasi Sistem

Dalam kegiatan implementasi sistem, dilakukan proses integrasi komponen backend sebagai inti dari fungsionalitas Sistem Informasi Pertanian Sumbawa (SiTAWA) yang berbasis website. Pengembangan backend difokuskan pada penyusunan layanan API yang dibangun menggunakan *framework Laravel*, yang berfungsi untuk menangani manajemen data pengaduan dari petani serta pelaporan hasil panen yang dikirim oleh setiap kecamatan. Sistem ini dirancang untuk menyediakan mekanisme digital yang efisien dalam menangani interaksi dua arah antara petani dan Dinas Pertanian Kabupaten Sumbawa. Dengan arsitektur backend yang terstruktur, sistem diharapkan mampu meningkatkan keteraturan, kecepatan pemrosesan, dan transparansi dalam pengelolaan data pertanian. Proses implementasi ini juga mencakup koneksi dengan *Supabase* sebagai penyedia layanan basis data untuk mendukung pengolahan dan penyimpanan data secara real-time dan terpusat.

a. Perancangan Database Server Supabase



Gambar 2. Pembuatan Projek pada Supabase

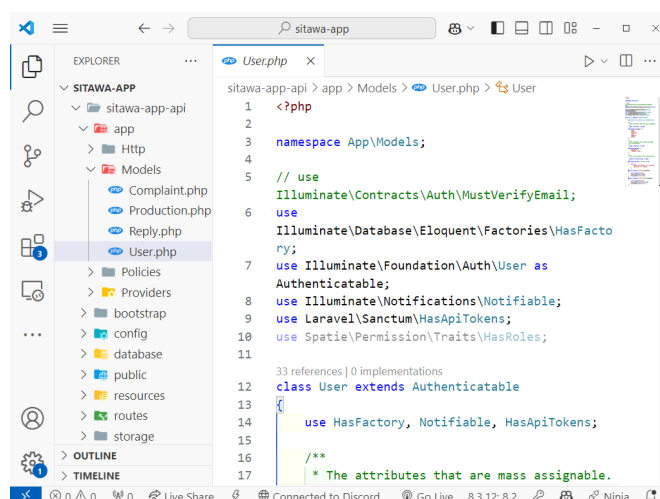
Pada Gambar 2 diperlihatkan proyek yang sudah dibuat untuk menyimpan *database website* SiTAWA (sitawa). Disini kita akan memanfaatkan server yang dimiliki oleh supabase untuk menyimpan data yang akan diolah di kedua aplikasi ini. Setelah itu sambungkan proyek *database* yang sudah dibuat ke proyek aplikasi.

```
1 DB_CONNECTION=pgsql
2 DB_URL=postgresql://postgres.sxsgoihulviciwqzqrj:47tIy63xIUBiDRU6@aws-0-us-east-1.pooler.supabase.com:6543/postgres
```

Gambar 3. Koneksi Database

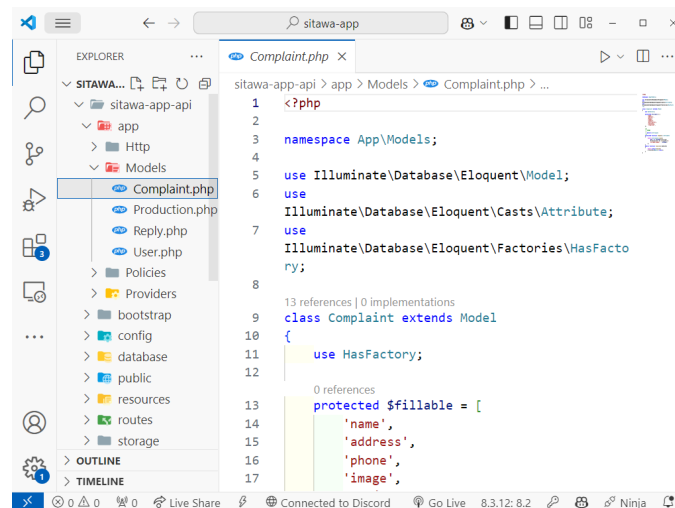
Koneksi database pada laravel dilakukan dengan mengisi “DB_CONNECTION” dan “DB_URL” pada file “.env”, dalam “DB_CONNECTION” diberikan nilai berupa jenis sistem manajemen basis data relasional, penulis menggunakan *postgresql* untuk jenisnya. Setelah itu penulis membuat koneksi dengan *server supabase* yang diisi dalam “DB_URL”.

b. Perancangan Model dan Controller



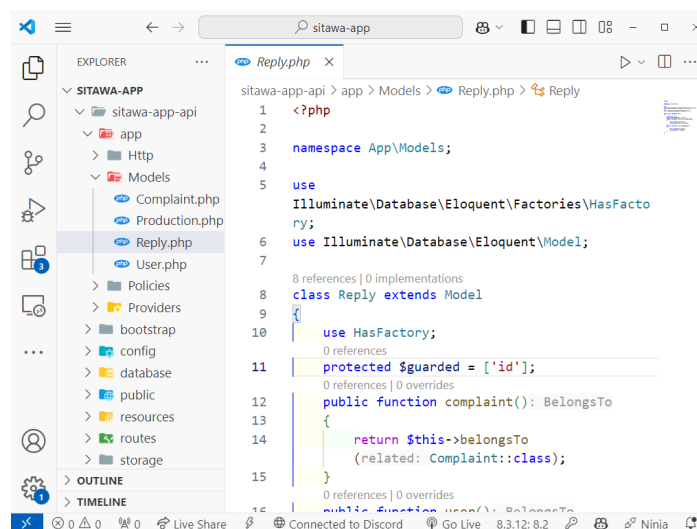
Gambar 4. Class Model User

Pada Gambar 4 di atas adalah representasi dari tabel pengguna dalam database yang digunakan oleh Laravel untuk mengelola data pengguna, autentikasi, dan interaksi terkait. Atribut “\$fillable” menentukan kolom yang dapat diisi secara langsung sedangkan “\$hidden” melindungi informasi sensitif seperti kata sandi dari ditampilkan dalam *output* JSON. Model ini juga mendefinisikan relasi *hasMany* dengan entitas *Reply* dan *Production*, yang berarti setiap pengguna dapat memiliki banyak balasan dan hasil produksi. Tujuan dibuatnya model *User* ini adalah untuk menyediakan struktur yang jelas dalam mengelola data pengguna, menjaga keamanan informasi, serta memungkinkan interaksi yang lebih terstruktur dan efisien menggunakan Laravel Eloquent ORM.



Gambar 5. Class Model Pengaduan

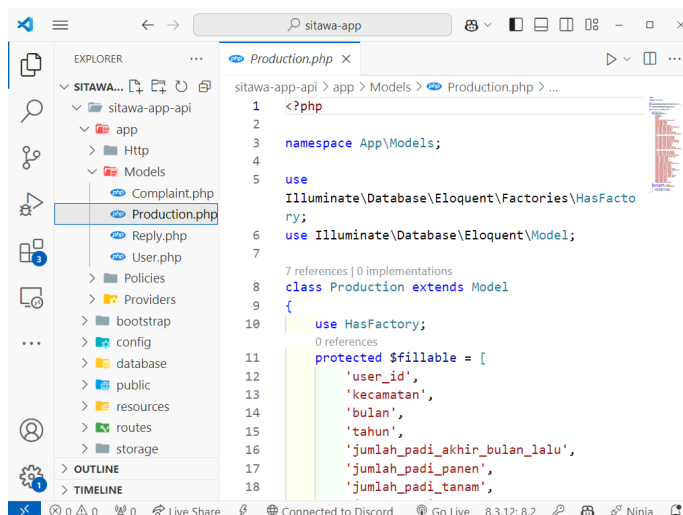
Pada Gambar 5 di atas adalah representasi dari entitas pengaduan dalam aplikasi Laravel, yang digunakan untuk mengelola data terkait pengaduan yang diajukan oleh pengguna. Model ini memiliki atribut “\$fillable” yang menunjukkan kolom-kolom yang dapat diisi secara langsung. Fungsi “image()” dalam model ini digunakan untuk membangun URL akses gambar yang tersimpan di direktori “/storage/image/”. Selain itu, model ini mendefinisikan relasi *hasOne* dengan entitas *Reply*, yang berarti satu pengaduan hanya dapat memiliki satu balasan. Tujuan dibuatnya model *Complaint* ini adalah untuk memfasilitasi pengelolaan data pengaduan secara terstruktur, mempermudah penyimpanan dan akses gambar yang terkait dengan pengaduan, serta mengatur hubungan dengan balasan dalam sistem Laravel.



Gambar 6. Class Model Balasan

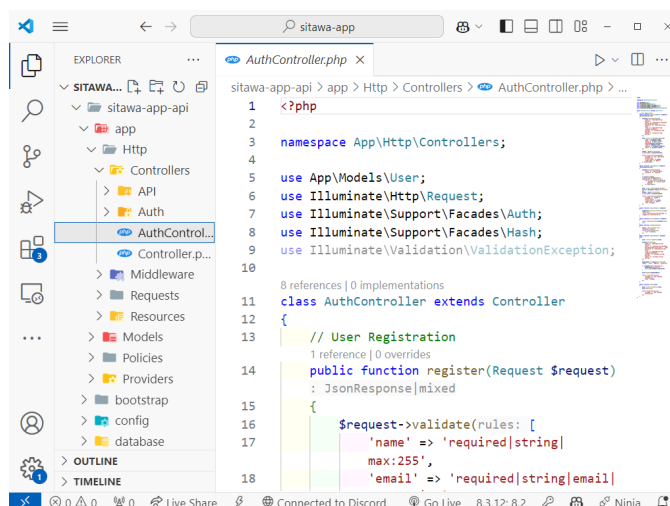
Pada Gambar 6 di atas adalah representasi dari entitas balasan dalam aplikasi Laravel, yang digunakan untuk mengelola data balasan terhadap pengaduan. Model ini menggunakan atribut “\$guarded” dengan kolom “id”, yang berarti semua kolom lain dapat diisi kecuali “id”, yang dilindungi dari pengisian massal. Relasi yang didefinisikan dalam model ini meliputi *belongsTo* dengan entitas *Complaint* dan *User*, yang menunjukkan bahwa setiap balasan terkait dengan satu pengaduan dan satu pengguna yang memberikan balasan. Tujuan dibuatnya model *Reply* ini adalah untuk memfasilitasi pengelolaan data balasan secara terstruktur, mengatur hubungan antara balasan dengan pengaduan dan pengguna, serta

memungkinkan Laravel Eloquent ORM mengelola interaksi antara balasan, pengaduan, dan pengguna dengan lebih efisien dalam sistem.



Gambar 7. Class Model Hasil Produksi

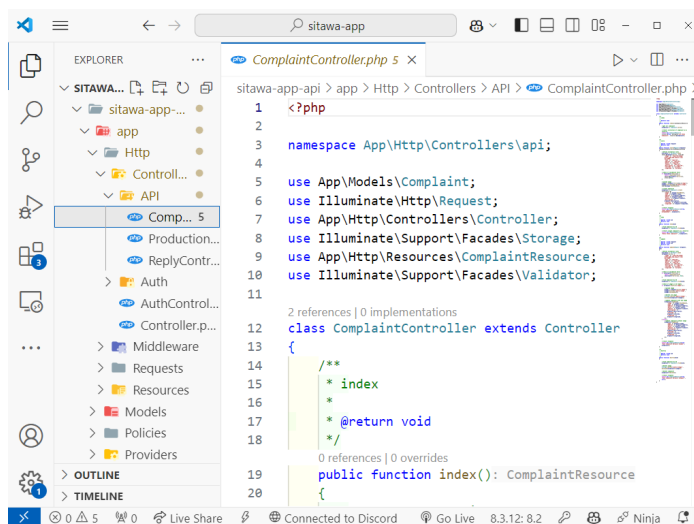
Pada Gambar 7 di atas adalah representasi dari entitas hasil produksi dalam aplikasi Laravel, yang berfungsi untuk mengelola data terkait hasil produksi yang dibuat oleh pengguna. Model ini menggunakan atribut “\$guarded” dengan kolom “id”, yang berarti semua kolom selain “id” dapat diisi secara massal, melindungi kolom “id” dari perubahan tidak sengaja. Relasi yang didefinisikan adalah *belongsTo* dengan entitas *User*, yang menunjukkan bahwa setiap hasil produksi dimiliki oleh satu pengguna. Tujuan dibuatnya model *Production* ini adalah untuk memfasilitasi pengelolaan data hasil produksi dalam aplikasi, mengatur hubungan antara hasil produksi dan pengguna yang membuatnya, serta memungkinkan interaksi yang lebih terstruktur dan efisien menggunakan Laravel Eloquent ORM.



Gambar 8. Class Controller User

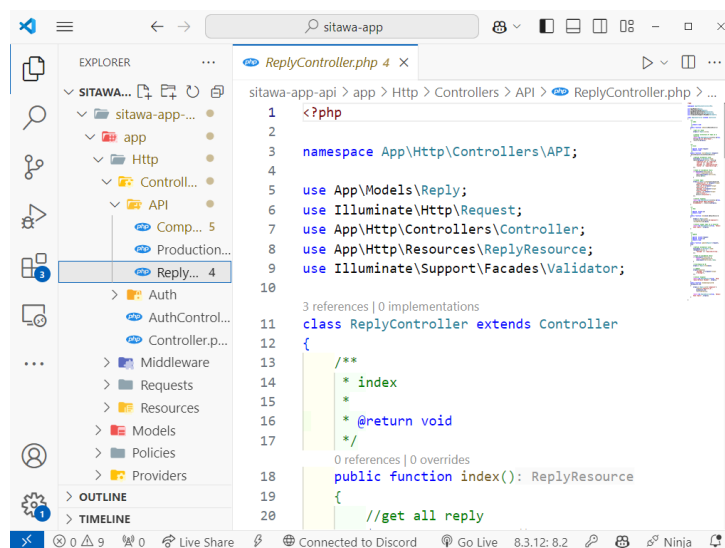
Pada Gambar 8 di atas adalah kontroler yang berfungsi untuk mengelola proses autentikasi dan manajemen pengguna dalam aplikasi Laravel. Kontroler ini mencakup beberapa metode penting, seperti pendaftaran pengguna (*register*), *login*, *logout*, menampilkan profil pengguna yang sedang *login*, mengambil daftar pengguna dengan peran tertentu (misalnya kecamatan), memperbarui data pengguna, dan menghapus pengguna. Setiap fungsi ini dilengkapi dengan validasi data untuk memastikan integritas input yang

diterima. Tujuan dibuatnya kontroler ini adalah untuk mengatur alur autentikasi dan otorisasi, mengelola token autentikasi menggunakan *Laravel Sanctum* untuk akses API, dan memberikan *endpoint* yang dapat digunakan oleh aplikasi *frontend* untuk berinteraksi dengan sistem manajemen pengguna secara aman dan terstruktur.



Gambar 9. Class Controller Pengaduan

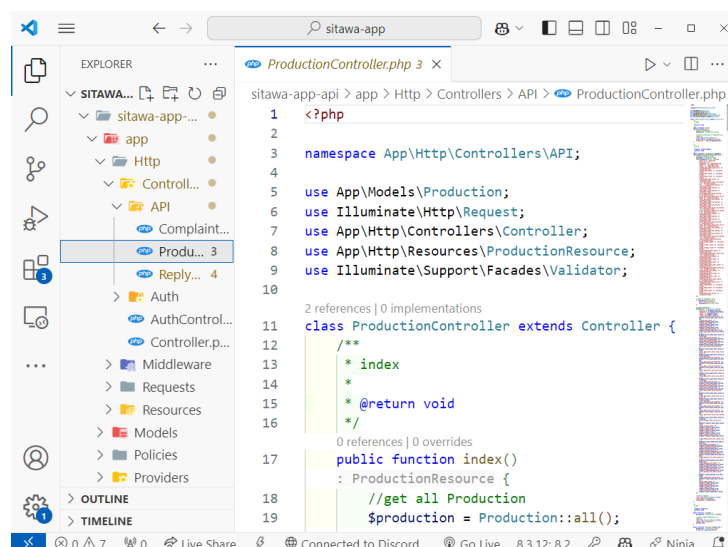
Pada Gambar 9 di atas adalah kontroler yang bertugas mengelola data pengaduan (*complaint*) dalam aplikasi berbasis Laravel. Tujuan utama dibuatnya kontroler ini adalah untuk menyediakan *endpoint* API yang memungkinkan pengguna atau aplikasi *frontend* untuk melakukan operasi CRUD (*Create, Read, Update, Delete*) terkait data pengaduan. Fungsinya meliputi menampilkan daftar semua pengaduan (*index*), menambah pengaduan baru dengan validasi dan unggahan gambar (*store*), menampilkan detail pengaduan tertentu (*show*), memperbarui data pengaduan termasuk pengelolaan file gambar (*update*), dan menghapus pengaduan serta file terkait dari penyimpanan (*destroy*). Kontroler ini memastikan pengelolaan data yang aman, terstruktur, dan mematuhi aturan validasi yang ditentukan, sambil mengelola penyimpanan file menggunakan *Laravel Storage*.



Gambar 10. Class Controller Balasan

Pada Gambar 10 di atas berfungsi sebagai pengelola data balasan (*reply*) dalam konteks pengaduan di aplikasi berbasis Laravel. Tujuan utama dari kontroler ini adalah untuk menyediakan *endpoint* API yang memungkinkan pengguna atau aplikasi *frontend* untuk

melakukan operasi CRUD (*Create, Read, Update*) terkait balasan terhadap pengaduan. Fungsinya mencakup menampilkan semua balasan (*index*), menambah balasan baru dengan validasi yang tepat (*store*), menampilkan detail balasan tertentu (*show*), dan memperbarui konten balasan yang sudah ada (*update*). Selain itu, kontroler ini juga menyediakan kemampuan untuk mengambil balasan berdasarkan ID pengaduan (*showByComplaintId*), yang mempermudah pengguna dalam melihat balasan yang relevan. Dengan pendekatan ini, *ReplyController* memastikan pengelolaan data yang terstruktur dan mematuhi aturan validasi yang ditetapkan, serta memberikan respon yang konsisten dalam bentuk sumber daya (*resource*).



Gambar 11. Class Controller Hasil Produksi

Pada Gambar 11 di atas berfungsi sebagai pengelola data hasil produksi padi dalam aplikasi berbasis Laravel. Kontroler ini menyediakan serangkaian *endpoint* API yang memungkinkan pengguna untuk melakukan operasi CRUD (*Create, Read, Update, Delete*) pada data produksi, termasuk menambahkan data hasil produksi baru, menampilkan semua data produksi, melihat detail dari data produksi tertentu, dan menghapus data produksi yang tidak diperlukan. Dengan fitur *showByUserId*, kontroler ini juga memungkinkan pengguna untuk mengambil data produksi berdasarkan ID pengguna, yang meningkatkan kemudahan akses informasi terkait hasil produksi individu. Tujuan dari pengembangan *ProductionController* adalah untuk memastikan bahwa data hasil produksi dapat dikelola secara efektif, serta memberikan validasi yang tepat pada data yang dimasukkan untuk menjaga integritas informasi dalam sistem.

3.2. Tantangan Implementasi dan Solusi Teknis

Selama proses implementasi backend SiTAWA, terdapat beberapa tantangan teknis yang harus dihadapi. Salah satu hambatan utama adalah integrasi dengan Supabase sebagai basis data dan layanan autentikasi. Pada tahap awal, struktur data bawaan Supabase kurang sesuai dengan kebutuhan sistem pelaporan yang kompleks dan multi-instansi, sehingga diperlukan penyesuaian skema database serta pembuatan fungsi khusus (*custom API logic*) untuk menangani relasi antar entitas secara lebih fleksibel.

Masalah lainnya terkait keamanan API, terutama dalam memastikan bahwa hanya pengguna yang terverifikasi dapat mengakses dan memodifikasi data sensitif. Untuk mengatasi hal ini, tim pengembang menerapkan *token-based authentication* menggunakan *JSON Web Token* (JWT), serta pembatasan akses berbasis peran (*role-based access control*). Audit log juga ditambahkan untuk merekam setiap aktivitas penting dalam sistem.

Selain itu, keterbatasan konektivitas internet di beberapa kecamatan menjadi kendala dalam uji coba awal. Solusi sementara yang diterapkan adalah penyimpanan data sementara secara lokal dengan sinkronisasi otomatis saat koneksi tersedia.

Pendekatan iteratif dan komunikasi rutin dengan pihak Dinas Kominfo dan Dinas Pertanian menjadi kunci dalam menyelesaikan tantangan-tantangan tersebut dan memastikan sistem berjalan stabil menjelang implementasi skala penuh.

3.3. Pangujian Sistem

Sistem yang dikembangkan diuji menggunakan metode *black-box*. *Blackbox testing* adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas sistem tanpa mengetahui struktur internal kode program. Pengujian dilakukan berdasarkan input dan output, dengan tujuan memastikan bahwa sistem bekerja sesuai dengan kebutuhan pengguna dan spesifikasi yang telah ditentukan (Fikri et al., 2024; Oktavian Eka & Sapriadi, 2024). Berikut hasil pengujian yang akan ditampilkan pada Tabel 1 yang sudah dilakukan.

Tabel 1. Hasil Pengujian Menggunakan Black Box Testing

No	Pengujian	Case	Output (diharapkan)	Hasil	Kesimpulan
1	Membuat data pengaduan	Memasukkan <i>value</i> Nama, noHP, Alamat, Gambar, Subjek, dan Keterangan	Menampilkan data baru yang sudah dibuat	Sesuai	<i>Valid</i>
2	Mengambil data pengaduan	Memasuki URL “api/complaints”	Menampilkan data	Sesuai	<i>Valid</i>
3	Mengambil data pengaduan dengan parameter <i>id</i>	Memasuki URL “api/complaints/{id}”	Menampilkan data yang sesuai dengan <i>id</i> terkait	Sesuai	<i>Valid</i>
4	Menghapus data pengaduan	Memasukkan <i>value</i> Id yang akan dihapus	Menampilkan pesan bahwa data telah dihapus	Sesuai	<i>Valid</i>
5	Membuat data balasan	Memasukkan <i>value</i> Id User, Id Complaint, Status, dan Pesan Balasan	Menampilkan data baru yang sudah dibuat	Sesuai	<i>Valid</i>
6	Mengambil data balasan dengan parameter Id pengaduan	Memasuki URL “api/repliesByComplaintId/{id}”	Menampilkan data sesuai dengan id terkait	Sesuai	<i>Valid</i>
7	Meng-update data Balasan yang sudah ada	Memasukkan <i>value</i> Id User, Id Complaint, Status, dan Pesan Balasan	Menampilkan data lama yang sudah diperbarui	Sesuai	<i>Valid</i>
8	Membuat data hasil produksi	Memasukkan <i>value</i> keterangan	Menampilkan data baru yang sudah dibuat	Sesuai	<i>Valid</i>
9	Mengambil data hasil Produksi	Memasuki URL “api/productions”	Menampilkan data	Sesuai	<i>Valid</i>
10	Mengambil data hasil produksi dengan	Memasuki URL “api/productions/{id}”	Menampilkan data sesuai dengan id terkait	Sesuai	<i>Valid</i>

	parameter id				
11	Menghapus data hasil produksi	Memasukkan <i>value</i> Id yang akan dihapus	Menampilkan pesan bahwa data telah dihapus	Sesuai	<i>Valid</i>
12	Menambah data user kecamatan	Memasukkan <i>value</i> Nama, <i>Email</i> , NoHP, Alamat, <i>Password</i> , dan <i>Role</i>	Menampilkan data baru yang sudah di buat	Sesuai	<i>Valid</i>
13	Melakukan <i>login user</i>	Memasukkan <i>value email</i> dan <i>password</i>	Menampilkan data <i>user</i> dan token	Sesuai	<i>Valid</i>
14	Melakukan <i>logout</i>	Memasukkan token <i>user</i> yang telah <i>login</i>	Menampilkan pesan bahwa user berhasil <i>logout</i>	Sesuai	<i>Valid</i>
15	Mengambil data <i>user</i> kecamatan	Memasuki URL “api/user”	Menampilkan data lama yang sudah diperbarui	Sesuai	<i>Valid</i>

Tabel 1 terlihat hasil dari pengujian menggunakan metode *blackbox* yang menghasilkan nilai valid. Hal ini menunjukkan bahwa semua proses yang ada pada aplikasi dapat berjalan dengan baik dan dapat digunakan.

4. Kesimpulan

Berdasarkan hasil pengerjaan *backend* website SiTAWA, dapat disimpulkan bahwa pengembangan backend telah diselesaikan sesuai dengan hasil diskusi awal bersama pihak Dinas Kominfotik dan Dinas Pertanian. Pengelolaan data pada sistem ini disederhanakan melalui API yang dirancang sesuai dengan fungsinya, sehingga mampu mendukung kebutuhan masing-masing instansi secara lebih efisien dan efektif. Namun demikian, penelitian ini masih memiliki keterbatasan pada integrasi dengan sistem eksternal seperti *e-Government regional* atau sistem pelaporan berbasis SMS yang masih digunakan sebagian petani. Untuk pengembangan selanjutnya, disarankan agar dilakukan perluasan modul integrasi lintas instansi dan peningkatan keamanan API melalui autentikasi berbasis token. Rencana implementasi sistem mencakup pelatihan bagi operator kecamatan, sosialisasi kepada kelompok tani, serta uji coba sistem secara bertahap di beberapa kecamatan sebelum diterapkan secara menyeluruh di seluruh wilayah Kabupaten Sumbawa.

Ucapan Terima Kasih

Ucapan terima kasih kami panjatkan kepada Allah SWT yang telah memberikan kekuatan, hidayah, dan rahmat-Nya sehingga penelitian ini dapat diselesaikan dengan baik. Kami juga menyampaikan terima kasih kepada berbagai pihak yang telah memberikan dukungan, khususnya kepada Dinas Komunikasi, Informatika dan Statistik (Diskominfotik) Kabupaten Sumbawa serta Kepala Bidang Dinas Pertanian Kabupaten Sumbawa atas kesempatan dan dukungan yang diberikan, sehingga proses penelitian ini dapat berjalan dengan lancar dan memberikan manfaat bagi kami.

Referensi

Ahmad, F., Irfan, P., & Abidin, Z. (2019). Penerapan Metode Fuzzy Tsukamoto untuk Menentukan Kelayakan Peminjaman pada Koperasi. *Jurnal Bumigora Information Technology (BITE)*, 1(2), 102–109. <https://doi.org/10.30812/BITE.V1I2.533>
Asrin, F., & Utami, G. V. (2024). Rancang Bangun Website SD Negeri 14 Pontianak Kota Dengan Pengujian Metode Black box. *Sains, Aplikasi, Komputasi Dan Teknologi*

- Informasi*, 5(1). <https://doi.org/10.30872/jsakti.v5i1.10301>
- Bournaris, T. (2020). Evaluation of e-government web portals: The case of agricultural e-government services in Greece. *Agronomy*, 10(7). <https://doi.org/10.3390/agronomy10070932>
- Fikri, R. R. N., Indera, I., Rahardi, A., & Agus, I. (2024). Pengujian Blackbox pada Sistem Informasi Komunitas Pecinta Kucing di Bandar Lampung. *Jurnal Teknik*, 18(1).
- Khoirudda'I Hermawan, M., Pandu Kusum, A., & Febrinita, F. (2022). Perancangan Sistem Pengelolaan Keuangan Hasil Panen Pertanian di Desa Sidodadi Kabupaten Blitar. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2). <https://doi.org/10.36040/jati.v6i2.5716>
- Mushi, G. E., Serugendo, G. D. M., & Burgi, P. Y. (2022). Digital Technology and Services for Sustainable Agriculture in Tanzania: A Literature Review. In *Sustainability (Switzerland)* (Vol. 14, Issue 4). <https://doi.org/10.3390/su14042415>
- Neves, F., & Silva, P. (2021). E-government in local governments' websites. *Revista Catarinense Da Ciência Contábil*, 20, e3160. <https://doi.org/10.16930/2237-7662202131602>
- Oktavian Eka, K., & Sapriadi. (2024). Sistem Informasi Pelaporan Kendaraan Dan Alat Berat Menggunakan Metode Rapid Application Development (Studi Kasus: PT. Andalas Karya Mulia). *Jurnal Testing Dan Implementasi Sistem Informasi*, 2(1).
- Wijaya, Y. D., & Astuti, M. W. (2021). Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan PT. Inka (Persero) Berbasis Equivalence Partitions. *Jurnal Digital Teknologi Informasi*, 4(1), 22–26. <https://doi.org/10.32502/DIGITAL.V4I1.3163>

[This page intentionally left blank.]