

Implementasi Metode *Waterfall* dan SUS pada Pengembangan *Website* Manajemen Peternakan Ayam *Broiler*

Ulfa Emi Rahmawati¹, Mohamad Al-Kahfi²

^{1,2} Jurusan Teknologi Informasi – Politeknik Negeri Jember

Jl. Mastrip, Krajan Timur, Sumbersari, Kabupaten Jember, 68121, Indonesia

ulfaemi@polije.ac.id, alkahfii2018@gmail.com

Abstrak— Peternakan ayam *broiler* adalah usaha budidaya ayam yang khusus dimanfaatkan dagingnya dalam waktu yang relatif singkat. Banyak peternak menghadapi permasalahan dalam pengelolaan kandang, pemantauan kesehatan ayam, maupun manajemen pakan. Tidak sedikit peternak juga mengeluhkan kesulitan dalam memantau pertumbuhan ternak dan profit mereka. Hal ini bisa membuat kualitas hasil produksi dan penghasilan peternak menurun. Penelitian ini berfokus pada pengembangan *website* manajemen peternakan ayam *broiler* dengan mengimplementasikan model *waterfall* karena memiliki tahapan pengembangan sistem yang jelas dan terdokumentasi sesuai prosedur berurutan. Pengujian yang dilakukan pada *website* menggunakan *System Usability Scale* (SUS) untuk mendapatkan hasil yang optimal dengan jumlah sampel yang terbatas. *Website* yang dikembangkan dalam penelitian ini dilengkapi fitur utama untuk mengatasi masalah yang dihadapi para peternak ayam *broiler*, yaitu manajemen kandang, manajemen populasi ayam berdasarkan kesehatannya, manajemen pakan, manajemen keuangan, *dashboard* visualisasi data, serta forum diskusi yang memudahkan peternak dalam bertukar informasi. Hasil penelitian telah memberikan dampak positif terhadap manajemen peternakan ayam *broiler*, sehingga mampu meningkatkan produktivitas ayam *broiler* dan memaksimalkan pendapatan peternak. Skor SUS yang diperoleh yakni 74,5 yang artinya *website* ini masuk ke dalam kategori cukup baik dalam hal kemudahan penggunaan. Penelitian selanjutnya dapat dilakukan peningkatan pada beberapa aspek *website*, terutama pada desain antarmuka dan kenyamanan *user* saat berinteraksi dengan *website*, sehingga dapat lebih meningkatkan efisiensi operasional manajemen peternakan ayam *broiler*.

Kata Kunci— Aplikasi *Website*, *Framework Laravel*, Manajemen Peternakan, Metode *Waterfall*, Peternakan Ayam *Broiler*, *System Usability Scale*

Abstract— *Broiler chicken farming is a chicken farming business specifically for meat production within a relatively short period of time. Many farmers face challenges in coop management, chicken health monitoring, and feed management. Many farmers also complain of difficulties in monitoring their livestock growth and profits. This can lead to decreased production quality and income. This research focuses on developing a broiler chicken farm management website using the waterfall model because it has clear and documented system development stages according to sequential procedures. Testing of the website was conducted using the System Usability Scale (SUS) to obtain optimal results with a limited sample size. The website developed in this research is equipped with key features to address the problems faced by broiler chicken farmers,*

namely coop management, chicken population management based on health, feed management, financial management, a data visualization dashboard, and a discussion forum that facilitates farmer information exchange. The research results have positively impacted broiler chicken farm management, thereby increasing broiler productivity and maximizing farmer income. The SUS score obtained was 74.5, indicating that the website falls into the fairly good category in terms of ease of use. Further research can be conducted to improve several aspects of the website, particularly the interface design and user experience when interacting with the website, thereby further enhancing the operational efficiency of broiler chicken farm management.

Keywords— *Website Application, Laravel Framework, Farm Management, Waterfall Method, Broiler Chicken Farm, System Usability Scale*

I. PENDAHULUAN

Keinginan masyarakat akan informasi yang cepat, tepat, dan mudah menjadi satu persyaratan yang wajib dipenuhi [1]. Hal ini juga berlaku di industri perunggasan di berbagai wilayah di Indonesia. Salah satunya industri peternakan ayam ras *broiler* [2]. Saat ini, permasalahan yang sering dihadapi para peternak ayam *broiler* meliputi pengelolaan kandang, pemantauan kesehatan ternak, serta manajemen pakan ternak [3].

Penelitian terdahulu menunjukkan bahwa *website* dirancang untuk memantau pertumbuhan ayam *broiler* melalui internet, dengan menggabungkan data dari berbagai peternak yang bekerja sama [1]. Namun, sistem ini belum memiliki fitur tambahan seperti mencatat populasi ayam berdasarkan kesehatannya, mengelola pakan ayam, membuat laporan keuangan, atau menganalisis data yang dapat membantu pengambilan keputusan secara lebih baik. Penelitian serupa memaparkan bahwa *website* yang dirancang berfokus pada peningkatan efisiensi operasional dan pengelolaan sumber daya di bidang peternakan [4]. Sistem ini belum menyediakan fitur tambahan seperti analisis pertumbuhan ternak serta integrasi pelaporan keuangan secara otomatis. Penelitian lain menyatakan bahwa *website* dibuat untuk mengelola berbagai data peternakan, seperti data jumlah ayam, data penjualan, data kematian ayam, dan biaya yang digunakan, seperti biaya pakan, biaya perawatan ayam, serta biaya perawatan kandang [5]. Sistem ini masih kurang dalam beberapa fitur lanjutan, seperti visualisasi data dan analisis tren pertumbuhan atau kematian ayam.

Jika manajemen ternak kurang baik, maka kualitas produksi

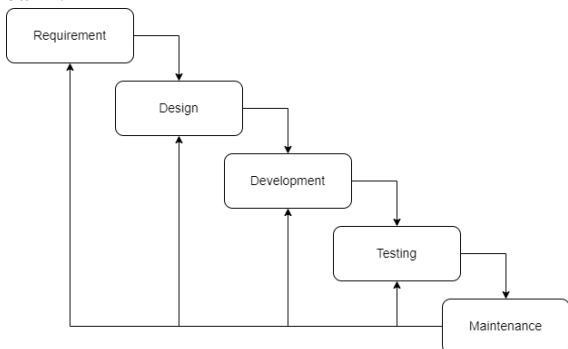
akan menurun dan risiko terjangkit penyakit serta kematian pada ayam akan meningkat serta pendapatan peternak juga menurun [4]. Dengan berkembangnya teknologi informasi, terdapat peluang besar untuk dapat menyediakan platform yang mendukung peternak sehingga dapat meningkatkan produktivitas dan efisiensi usaha [5].

Berdasarkan penelitian sebelumnya, solusi tepat untuk menyediakan pengelolaan manajemen kepada para peternak adalah dengan menggunakan teknologi informasi, terutama pengembangan *website* [5]. *Website* dapat digunakan sebagai media yang dapat diakses kapan dan dimana saja tanpa batasan waktu [6]. Maka dari itu, fokus utama dari penelitian ini adalah untuk mengembangkan *website* manajemen peternakan ayam *broiler* yang efisien dan efektif dengan beberapa fitur potensial untuk pengelolaan ternak ayam *broiler*. Beberapa fitur yang perlu dikembangkan lebih lanjut pada *website* manajemen peternakan ayam *broiler* yang akan dikembangkan yaitu fitur-fitur yang tidak hanya membantu mengoptimalkan operasional peternakan ayam *broiler*, tetapi juga menekankan pentingnya manajemen ternak termasuk pelaporan keuangan. Tujuannya adalah untuk meningkatkan keterampilan para peternak dalam mengelola usahanya sehingga para peternak lebih siap menghadapi tantangan dalam industri peternakan. Pengembangan *website* dalam penelitian ini menggunakan metode *waterfall* karena memiliki tahapan pengembangan sistem yang jelas serta terdokumentasi sesuai prosedur berurutan [7].

Pengujian pada *website* menggunakan *System Usability Scale (SUS)* karena mampu menunjukkan hasil yang optimal meskipun dengan jumlah sampel yang terbatas [8]. Hasil yang diperoleh dalam penelitian ini adalah sebuah *website* manajemen peternakan ayam *broiler* yang mudah diakses sehingga para peternak dapat mengelola usaha dengan lebih efisien sehingga dapat berkontribusi pada peningkatan produktivitas hewan ternak dan dapat memaksimalkan pendapatan peternak. Selain itu, juga membantu peternak untuk dapat melakukan pengambilan keputusan secara lebih baik dari analisis tren pertumbuhan atau kematian ayam dan integrasi manajemen pelaporan keuangan.

II. METODE PENELITIAN

Tahapan pengembangan *website* menggunakan metode *waterfall*. Metode ini menyediakan alur hidup pengembangan perangkat lunak secara berurutan [9], seperti tertera pada Gambar 1.



Gbr 1. Tahapan dalam Metode Waterfall [9]

Setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya untuk memastikan tidak ada tahapan yang terlewat [9]. Tahapan-tahapan tersebut adalah sebagai berikut:

A. Requirement

Tahapan ini berfokus untuk mengumpulkan data mengenai kebutuhan yang harus dipenuhi oleh sistem sesuai kebutuhan *user* mulai dari *functional requirement* dan *non-functional requirement* yang bersumber dari wawancara *end-user* serta studi literatur dari berbagai jurnal dan buku elektronik terkait.

B. Design

Tahapan ini berfokus untuk melakukan perancangan *database* yakni *Entity Relationship Diagram (ERD)* serta perancangan sistem meliputi *architecture system*, *flowchart*, dan *use case diagram* berdasarkan kebutuhan yang sudah dihimpun sebelumnya.

C. Development

Tahapan ini berguna untuk implementasi dari perancangan yang telah dibuat sebelumnya. Fokus utama pada tahap ini adalah pengimplementasian desain menjadi kode program yang dapat dijalankan. Proses *development* ini mencakup pembuatan *front-end website* menggunakan *HTML*, *CSS*, dan *JavaScript* serta *back-end* menggunakan *framework Laravel*. Untuk pengelolaan dan penyimpanan data secara efisien menggunakan *database MySQL*.

D. Testing

Tahapan ini berfokus untuk menguji sistem yang telah dikembangkan. Pengujian dilakukan dari sisi *user experience* dengan metode pengujian *System Usability Testing (SUS)*. Metode ini bertujuan mengukur seberapa baik sistem bisa digunakan dengan efektif, efisien, dan memuaskan bagi *end-user* [10]. *SUS* menggunakan kuisisioner yang terdiri dari 10 pertanyaan dengan skala *Likert 5* poin, dari sangat tidak setuju hingga sangat setuju [11]. Pada tahap ini, *user* diminta untuk menggunakan *website* secara langsung dan memberikan masukan melalui kuisisioner, kemudian hasil kuisisioner akan dihitung berapa nilai *usability* dari sistem tersebut.

E. Maintenance

Tahapan ini berfokus untuk memberikan dukungan pada *website* guna memastikan semua fitur dapat berjalan dengan baik dan setiap masalah yang mungkin muncul bisa segera ditangani.

III. HASIL DAN PEMBAHASAN

A. Requirement

Requirement dilakukan secara mendetail, mencakup dua kategori utama, yaitu *functional requirement* dan *non-functional requirement*. *Functional requirement* mencakup berbagai fungsi yang wajib terdapat dalam *website*. *Non-functional requirement* mencakup aspek yang berkaitan dengan bagaimana *website* bekerja. *Requirement website* manajemen peternakan ayam *broiler* seperti terlihat pada Tabel 1.

TABEL I
FUNCTIONAL REQUIREMENT DAN NON-FUNCTIONAL REQUIREMENT WEBSITE
MANAJEMEN PETERNAKAN AYAM BROILER

Functional Requirement	Non-Functional Requirement
Website harus memiliki akses umum ke halaman landing page tanpa memerlukan login.	Antarmuka website harus mudah dipahami dan mudah digunakan oleh user.
Website harus memiliki fitur autentikasi untuk login.	Website harus dapat digunakan melalui perangkat mobile maupun desktop.
Website harus memiliki dashboard yang berisi visualisasi data ringkasan, diantaranya ringkasan keuangan, jumlah kandang, ringkasan manajemen populasi ayam berdasarkan kesehatan ayam.	Website harus memberikan respons dengan cepat ketika digunakan.
Website harus memiliki fitur untuk manajemen keuangan, kandang, populasi ayam berdasarkan kesehatan ayam, dan pakan ayam, meliputi fungsi menambah, melihat, mengubah, dan menghapus data.	Forum diskusi harus dapat membantu komunikasi yang baik antar user.
Website harus memiliki fitur forum diskusi yang memungkinkan user merespons komentar atau topik yang dibuat peternak lain.	Website harus menggunakan Bahasa Indonesia yang mudah dimengerti.

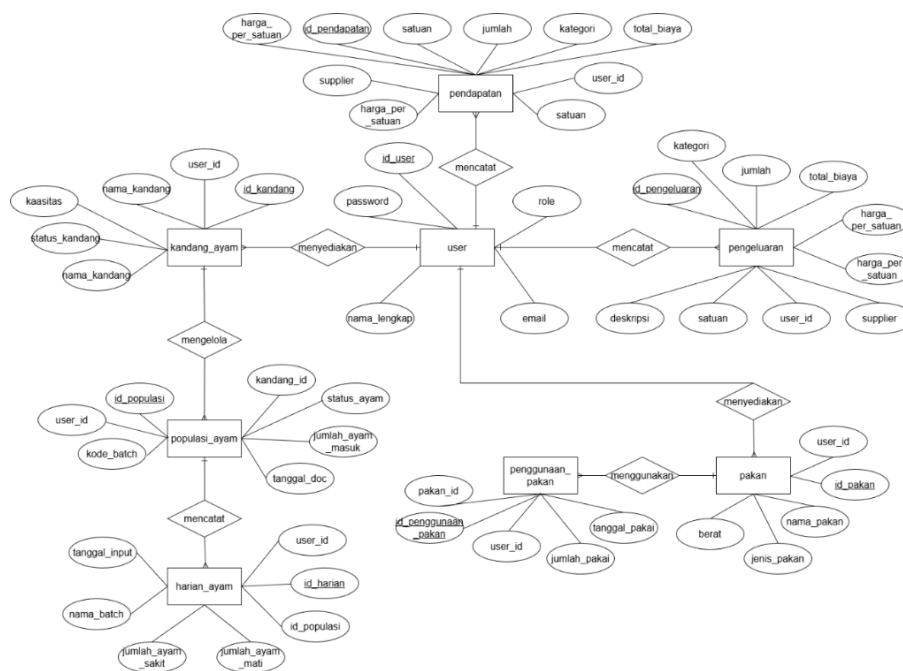
Functional Requirement	Non-Functional Requirement
Website harus memiliki fitur untuk mengelola profil user, diantaranya fungsi melihat, mengubah, dan menghapus profil.	

B. Design

Design dibuat supaya setiap bagian dalam website dirancang secara rinci dan dapat memenuhi functional requirement dan non-functional requirement yang sudah dirincikan pada Tabel 1.

1) Entity Relationship Diagram (ERD)

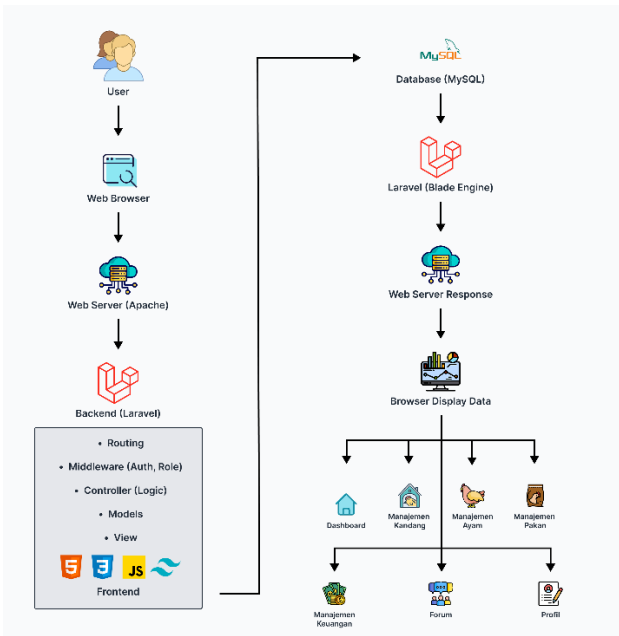
Desain Entity Relationship Diagram (ERD) menunjukkan bagaimana entitas-entitas dalam website saling berkaitan. Diagram ini mencakup berbagai entitas. User dapat mengelola kandang ayam, seperti menambah kapasitas dan memperbarui status kandang. Populasi mencakup informasi tentang jumlah ayam yang masuk, sedangkan entitas harian mencakup data mengenai ayam yang sakit serta ayam yang mati. User mencatat manajemen keuangan pada entitas pendapatan dan pengeluaran. Fitur pakan memungkinkan user menambahkan informasi mengenai detail jenis pakan yang digunakan. Pencatatan ini berguna untuk memantau efisiensi dan kebutuhan pakan sehari-hari. Fitur forum diskusi memungkinkan user membuat topik, menambahkan komentar, serta memberi like pada topik atau komentar tertentu melalui fitur suka. Gambar 2 menunjukkan desain ERD dari website manajemen peternakan ayam broiler.



Gbr 2. Desain ERD Website Manajemen Peternakan Ayam Broiler

2) Architecture System

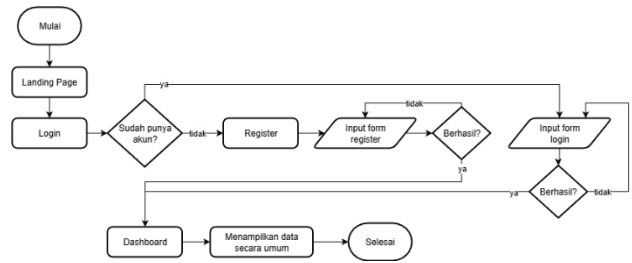
Architecture system memaparkan bagaimana website beroperasi. Proses dimulai ketika user membuka website melalui browser, kemudian mengirimkan permintaan HTTP ke server, lalu permintaan tersebut diteruskan ke web server. Permintaan diteruskan ke back-end Laravel, yang melakukan proses routing untuk memastikan URL sesuai dengan controller, kemudian memeriksa autentikasi login user melalui middleware. Controller kemudian menangani logika aplikasi seperti create, read, update, dan delete. Laravel mengakses database MySQL untuk menjalankan query. Front-end dikirimkan kembali ke browser supaya dapat ditampilkan kepada user. Gambar 3 merupakan architecture website manajemen peternakan ayam broiler.



Gbr 3. Architecture Website Manajemen Peternakan Ayam Broiler

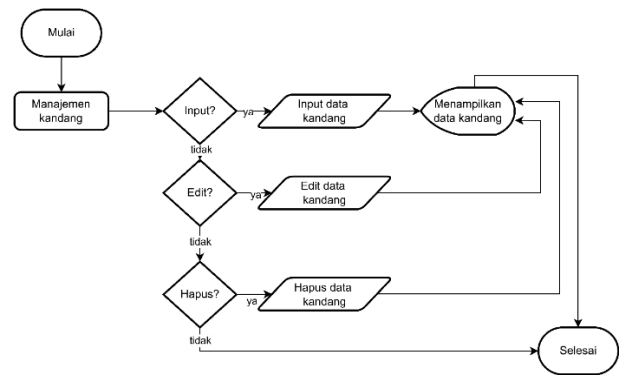
3) Flowchart

Flowchart pertama yakni proses login user. User mengakses landing page terlebih dahulu. Proses login dimulai dengan memeriksa apakah user sudah memiliki akun atau belum. User yang belum memiliki akun akan diarahkan ke halaman register, dimana user harus mengisi form yang tersedia. Sistem akan membawa user ke dashboard setelah proses register selesai. Jika user sudah memiliki akun, sistem akan menampilkan form login. Setelah username dan password dimasukkan, sistem akan melakukan autentikasi. Login yang berhasil akan mengarahkan user ke halaman dashboard. Flowchart login user terlihat pada Gambar 4.



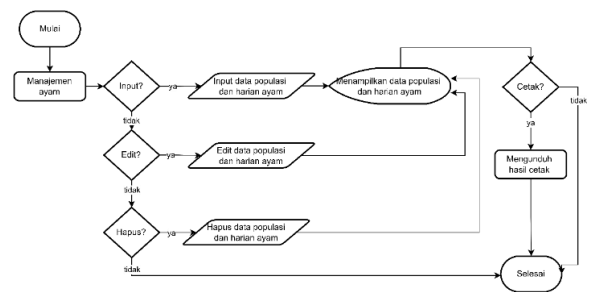
Gbr 4. Flowchart Login User

Flowchart kedua yakni proses manajemen kandang. Pada fitur ini menunjukkan user dapat melakukan manajemen kandang meliputi menginput, mengubah, melihat, dan menghapus data kandang. Proses ini dirancang agar mudah dipahami dan mendukung efisiensi dalam pengelolaan data kandang secara menyeluruh. Gambar 5 merupakan flowchart manajemen kandang.



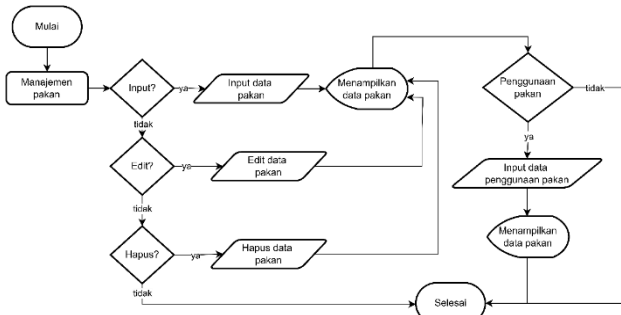
Gbr 5. Flowchart Manajemen Kandang

Flowchart ketiga yakni proses manajemen populasi ayam berdasarkan kesehatannya. Pada fitur ini menunjukkan user dapat melakukan manajemen data jumlah populasi dan data harian ayam berdasarkan kondisi kesehatannya meliputi menginput, mengubah, melihat, dan menghapus data ayam. Selain itu, terdapat opsi cetak laporan berupa pdf. Flowchart manajemen populasi ayam berdasarkan kesehatannya terlihat pada Gambar 6.



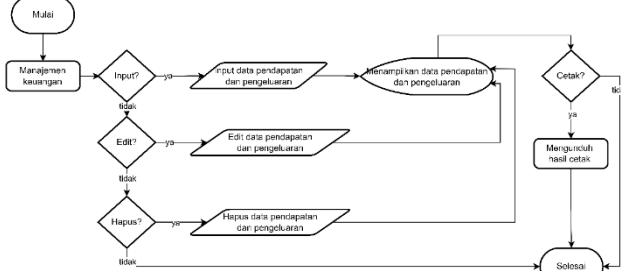
Gbr 6. Flowchart Manajemen Populasi Ayam Berdasarkan Kesehatannya

Flowchart keempat yakni proses manajemen pakan. Pada fitur ini menunjukkan user dapat melakukan manajemen data stok dan penggunaan pakan meliputi menginput, mengubah, melihat, dan menghapus data pakan. Gambar 7 merupakan flowchart manajemen pakan.



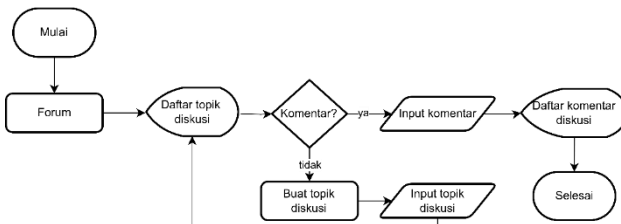
Gbr 7. Flowchart Manajemen Pakan

Flowchart kelima yakni proses manajemen keuangan. Pada fitur ini menunjukkan user dapat melakukan manajemen data pemasukan dan pengeluaran meliputi menginput, mengubah, melihat, dan menghapus data keuangan. Selain itu, terdapat opsi cetak laporan berupa pdf. Flowchart manajemen keuangan seperti terlihat pada Gambar 8.



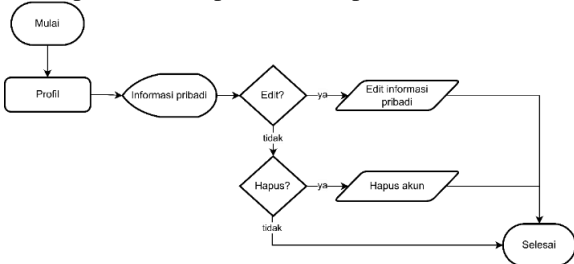
Gbr 8. Flowchart Manajemen Keuangan

Flowchart keenam yakni proses penggunaan fitur forum diskusi. Pada fitur ini menunjukkan user dapat melihat topik diskusi dari user lain, memberikan komentar pada topik yang sudah ada, serta membuat topik diskusi baru, Gambar 9 merupakan flowchart forum diskusi.



Gbr 9. Flowchart Forum Diskusi

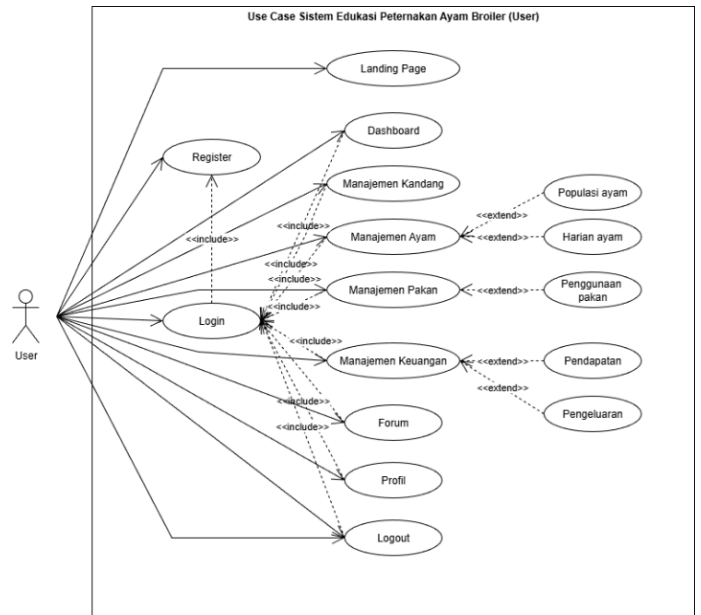
Flowchart ketujuh yakni proses penggunaan fitur profil user. Pada fitur ini menunjukkan user dapat melihat informasi pribadi, memperbarui informasi pribadi, serta menghapus akun. Flowchart profil user seperti terlihat pada Gambar 10.



Gbr 10. Flowchart Profil User

4) Use Case Diagram

Interaksi user dengan website manajemen peternakan ayam broiler dimulai dari landing page. User diberikan dua pilihan untuk mengakses fitur utama di dalam website, yaitu register atau login. Jika user belum memiliki akun, maka dapat menggunakan fitur register yang secara otomatis akan masuk ke proses login setelah registrasi berhasil, seperti yang terlihat dalam hubungan <<include>>. Setelah berhasil login, user akan diarahkan ke halaman dashboard, yang menjadi pusat pengaturan utama untuk mengakses semua fitur di dalam website. User dapat menggunakan berbagai fitur utama setelah login, yang ditunjukkan melalui hubungan <<include>> masing-masing fitur. Fitur-fitur tersebut meliputi manajemen kandang, manajemen populasi ayam berdasarkan kondisi kesehatannya, manajemen pakan, manajemen keuangan, forum diskusi, dan profil user. Beberapa fitur memiliki turunan ke fitur lain yang melekat pada fitur tersebut, yang ditunjukkan dengan hubungan <<extend>>. Gambar 11 menunjukkan use case diagram website manajemen peternakan ayam broiler.



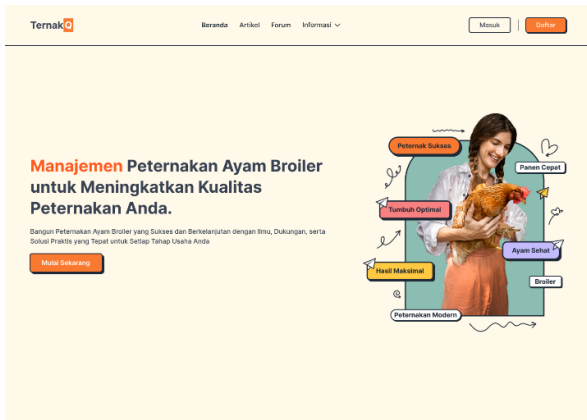
Gbr 11. Use Case Diagram Website Manajemen Peternakan Ayam Broiler

C. Development

Development website dilakukan setelah proses design selesai. Design yang sudah dibuat pada tahap sebelumnya diubah menjadi kode program yang dapat dijalankan. Development dilakukan di dua bagian, yaitu front-end dan back-end. Front-end dibuat untuk menampilkan antarmuka yang dapat diakses oleh user secara interaktif dan responsif, sedangkan back-end dikembangkan untuk mengelola logika website serta mengintegrasikan database yang sudah dirancang sebelumnya.

1) Halaman Landing Page

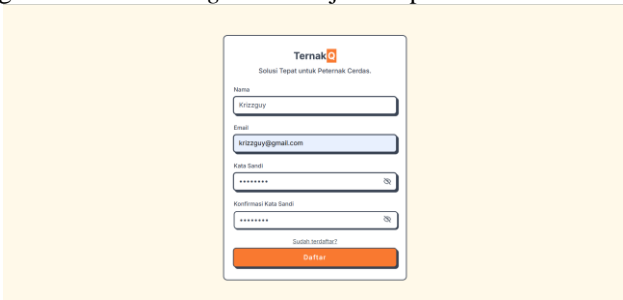
Halaman landing page merupakan halaman awal yang dapat diakses secara umum oleh user tanpa memerlukan login, seperti terlihat pada Gambar 12.



Gbr 12. Halaman Landing Page

2) Halaman Register

Halaman register berisi form pendaftaran bagi user baru untuk membuat akun di website manajemen peternakan ayam broiler, seperti terlihat pada Gambar 13. Kode program untuk algoritma halaman register ditunjukkan pada Gambar 14.



Gbr 12. Halaman Register

```
public function register(): void
{
    $validated = $this->validate([
        'name' => ['required', 'string', 'max:255', 'min:3'],
        'email' => ['required', 'string', 'lowercase', 'email', 'max:255', 'unique:User:email'],
        'password' => ['required', 'string', 'min:8', 'confirmed', Rules\Password::defaults()],
    ]);

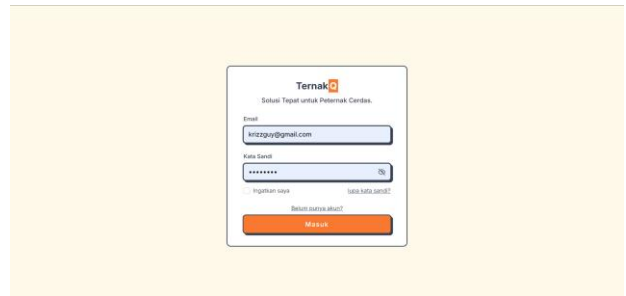
    [
        'name.required' => 'Nama harus diisi',
        'name.min' => 'Nama minimal 3 karakter',
        'email.required' => 'Email harus diisi',
        'email.email' => 'Format email tidak valid',
        'email.unique' => 'Email sudah terdaftar',
        'password.required' => 'Kata sandi harus diisi',
        'password.min' => 'Kata sandi minimal 8 karakter',
        'password.confirmed' => 'Konfirmasi kata sandi tidak cocok',
    ];

    $validated['password'] = Hash::make($validated['password']);
    event(new Registered($user = User::create($validated)));
    Auth::login($user);
    $this->redirect(RouteServiceProvider::HOME, navigate: true);
}
}
```

Gbr 13. Kode Program Halaman Register

3) Halaman Login

Halaman login digunakan oleh user yang telah memiliki akun untuk masuk kedalam website, pada halaman ini terdapat autentikasi user. Gambar 14 menyajikan halaman login dan Gambar 15 menyajikan kode program untuk algoritma login.



Gbr 14. Halaman Login

```
public function authenticate(): void
{
    $this->validate();
    $this->ensureIsNotRateLimited();
    $user = \App\Models\User::where('email', $this->email)->first();
    if (!$user) {
        RateLimiter::hit($this->throttleKey());

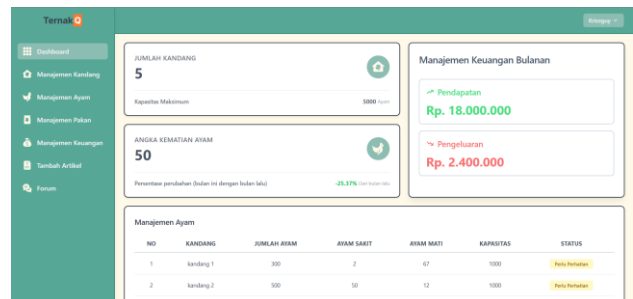
        throw ValidationException::withMessages([
            'form.email' => 'Email tidak terdaftar.',
        ]);
    }
    if (!Auth::attempt($this->only(['email', 'password']), $this->remember)) {
        RateLimiter::hit($this->throttleKey());

        throw ValidationException::withMessages([
            'form.password' => 'Kata sandi yang Anda masukkan salah.',
        ]);
    }
    RateLimiter::clear($this->throttleKey());
}
}
```

Gbr 15. Kode Program Halaman Login

4) Halaman Dashboard

Halaman dashboard menampilkan ringkasan data seperti manajemen kandang ayam, manajemen populasi ayam berdasarkan kesehatannya, dan informasi keuangan, seperti terlihat pada Gambar 16. Kode program untuk algoritma halaman dashboard ditunjukkan pada Gambar 17.



Gbr 16. Halaman Dashboard

```

class DashboardController extends Controller
{
    public function index(Request $request)
    {
        $userId = Auth::id();
        $totalKandangAyams = KandangAyam::where('status_kandang', 'Aktif')
            ->where('user_id', $userId)
            ->count();

        $totalCapacity = KandangAyam::where('status_kandang', 'Aktif')
            ->where('user_id', $userId)
            ->sum('kapasitas');

        $totalDeathsThisMonth = DB::table('harian_ayam')
            ->whereMonth('tanggal_input', Carbon::now()->month)
            ->whereYear('tanggal_input', Carbon::now()->year)
            ->whereIn('id_populasi', function ($query) use ($userId) {
                $query->select('id')->from('populasi_ayam')->where('user_id', $userId);
            })
            ->sum('jumlah_ayam_mati');

        $currentMonthDeaths = DB::table('harian_ayam')
            ->whereYear('tanggal_input', Carbon::now()->year)
            ->whereMonth('tanggal_input', Carbon::now()->month)
            ->whereIn('id_populasi', function ($query) use ($userId) {
                $query->select('id')->from('populasi_ayam')->where('user_id', $userId);
            })
            ->sum('jumlah_ayam_mati');

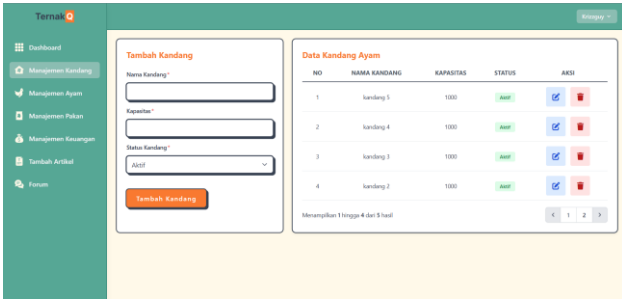
        $previousMonthDeaths = DB::table('harian_ayam')
            ->whereYear('tanggal_input', Carbon::now()->year)
            ->whereMonth('tanggal_input', Carbon::now()->subMonth()->month)
            ->whereIn('id_populasi', function ($query) use ($userId) {
                $query->select('id')->from('populasi_ayam')->where('user_id', $userId);
            })
            ->sum('jumlah_ayam_mati');
    }
}

```

Gbr 17. Kode Program Halaman Dashboard

5) Halaman Manajemen Kandang

Halaman manajemen kandang berisi tabel informasi data kandang dan form untuk menambah, melihat, mengubah dan menghapus data kandang, seperti terlihat pada Gambar 18. Kode program untuk algoritma halaman manajemen kandang ditunjukkan pada Gambar 19.



Gbr 18. Halaman Manajemen Kandang

```

class KandangAyamController extends Controller
{
    public function indexKandangManagement(Request $request)
    {
        try {
            $kandangPage = $request->get('kandang_page', 1);
            $kandang = KandangAyam::where('user_id', Auth::id())
                ->latest()
                ->paginate(4, ['*', 'kandang_page', $kandangPage]);
            return view('cage-management', compact('kandang'));
        } catch (\Exception $e) {
            Log::error('Gagal memuat data kandang: ' . $e->getMessage());
        }

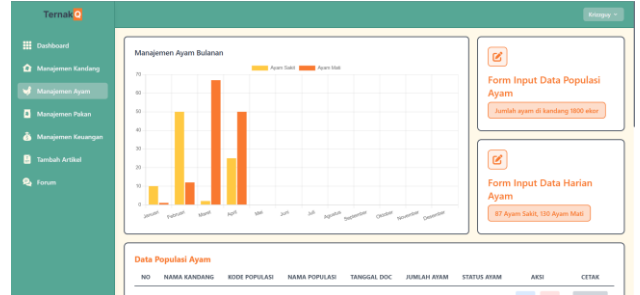
        return redirect()->route('cage-management')->with([
            'status' => 'error',
            'message' => 'Terjadi kesalahan saat memuat data kandang.'
        ]);
    }
}

```

Gbr 19. Kode Program Halaman Manajemen Kandang

6) Halaman Manajemen Populasi Ayam Berdasarkan Kesehatannya

Halaman manajemen populasi ayam berdasarkan kesehatannya berisi data populasi ayam berdasarkan kesehatannya dan data harian ayam berdasarkan kapasitas kandang. Selain itu, terdapat grafik harian ayam sakit dan mati berdasarkan bulan, seperti terlihat pada Gambar 20. Kode program untuk algoritma halaman manajemen populasi ayam berdasarkan kesehatannya ditunjukkan pada Gambar 21.



Gbr 20. Halaman Manajemen Populasi Ayam Berdasarkan Kesehatannya

```

class PopulasiHarianController extends Controller
{
    public function indexChickenManagement(Request $request)
    {
        try {
            $populasiPage = $request->get('populasi_page', 1);
            $harianPage = $request->get('harian_page', 1);
            $populasi = PopulasiAyam::with('kandang')
                ->where('user_id', Auth::id())
                ->latest()
                ->paginate(5, ['*', 'populasi_page', $populasiPage]);
            $harian = HarianAyam::whereHas('populasiAyam', function($query) {
                $query->where('user_id', Auth::id());
            })
                ->latest()
                ->paginate(5, ['*', 'harian_page', $harianPage]);

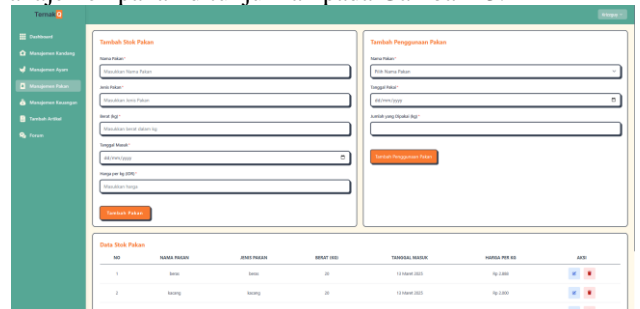
            $batches = PopulasiAyam::where('user_id', Auth::id())->get();
            $kandang = KandangAyam::where('status_kandang', 'Aktif')
                ->where('user_id', Auth::id())
                ->get();
            $userId = Auth::id();
            $monthlyData = DB::table('harian_ayam')
                ->join('populasi_ayam', 'harian_ayam.id_populasi', '=', 'populasi_ayam.id')
                ->whereIn('populasi_ayam.status_ayam', ['Proses', 'Siap Panen'])
                ->where('populasi_ayam.user_id', $userId)
                ->select(
                    DB::raw('MONTH(harian_ayam.tanggal_input) as month'),
                    DB::raw('SUM(harian_ayam.jumlah_ayam_sakit) as sick'),
                    DB::raw('SUM(harian_ayam.jumlah_ayam_mati) as dead')
                );
        }
    }
}

```

Gbr 21. Kode Program Halaman Manajemen Populasi Ayam Berdasarkan Kesehatannya

7) Halaman Manajemen Pakan

Halaman manajemen pakan berisi form untuk mengelola persediaan dan penggunaan pakan sehingga user dapat melacak arus masuk dan keluar pakan secara akurat, seperti terlihat pada Gambar 22. Kode program untuk algoritma halaman manajemen pakan ditunjukkan pada Gambar 23.



Gbr 22. Halaman Manajemen Pakan

```
class PakanController extends Controller
{
    public function indexPakan(Request $request)
    {
        try {
            $pakanPage = $request->get('pakan_page', 1);
            $pakan = Pakan::where('user_id', Auth::id())
                ->latest()
                ->paginate(5, ['*', 'pakan_page', $pakanPage]);

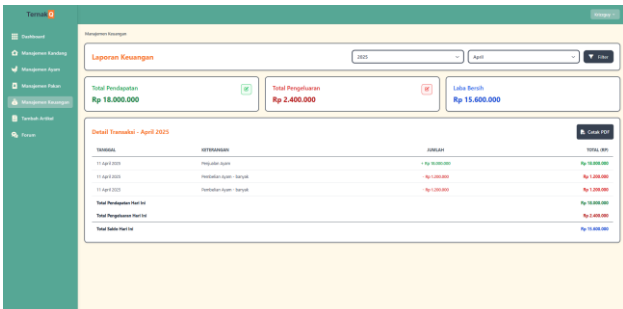
            return view('food-management', compact('pakan'));
        } catch (\Exception $e) {
            Log::error('Gagal memuat data pakan: ' . $e->getMessage());

            return redirect()->route('food-management')->with([
                'status' => 'error',
                'message' => 'Terjadi kesalahan saat memuat data pakan.',
            ]);
        }
    }
}
```

Gbr 23. Halaman Manajemen Pakan

8) Halaman Manajemen Keuangan

Halaman manajemen keuangan menampilkan laporan keuangan yang dapat dilihat berdasarkan bulan dan tahun. Selain itu juga menampilkan penghasilan, pengeluaran, dan keuntungan. seperti terlihat pada Gambar 24. Gambar 25 menyajikan kode program untuk algoritma penghasilan dan Gambar 26 menyajikan program untuk algoritma pengeluaran.



Gbr 24. Halaman Manajemen Keuangan

```
class PendapatanController extends Controller
{
    public function indexPendapatan(Request $request)
    {
        try {
            $pendapatanPage = $request->get('pendapatan_page', 1);

            $pendapatan = Pendapatan::where('user_id', Auth::id())
                ->orderBy('tanggal_transaksi', 'desc')
                ->paginate(10, ['*', 'pendapatan_page', $pendapatanPage]);

            return view('finance-management-income', compact('pendapatan'));
        } catch (\Exception $e) {
            Log::error('Gagal memuat data pendapatan: ' . $e->getMessage());

            return redirect()->route('finance-management-income')->with([
                'status' => 'error',
                'message' => 'Terjadi kesalahan saat memuat data pendapatan.',
            ]);
        }
    }
}
```

Gbr 25. Kode Program Penghasilan

```
class PengeluaranController extends Controller
{
    public function indexPengeluaran(Request $request)
    {
        try {
            $pengeluaranPage = $request->get('pengeluaran_page', 1);
            $pengeluaran = Pengeluaran::where('user_id', Auth::id())
                ->orderBy('tanggal_pembelian', 'desc')
                ->paginate(10, ['*', 'pengeluaran_page', $pengeluaranPage]);

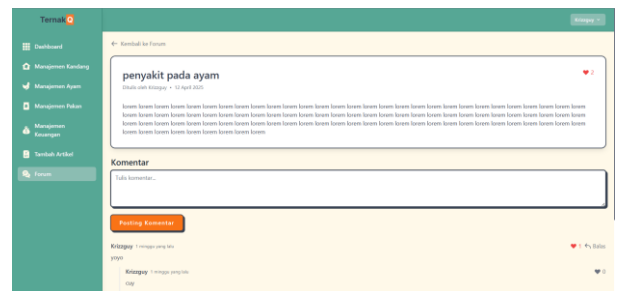
            return view('finance-management-outcome', compact('pengeluaran'));
        } catch (\Exception $e) {
            Log::error('Gagal memuat data pengeluaran: ' . $e->getMessage());

            return redirect()->route('finance-management-outcome')->with([
                'status' => 'error',
                'message' => 'Terjadi kesalahan saat memuat data pengeluaran.',
            ]);
        }
    }
}
```

Gbr 26. Kode Program Pengeluaran

9) Halaman Forum Diskusi

Halaman forum diskusi berfungsi sebagai tempat para peternak saling berinteraksi, semua topik yang dibuat oleh user ditampilkan dalam bentuk list. User dapat ikut berdiskusi atau menambahkan topik baru sesuai kebutuhan peternakan ayam broiler mereka, seperti terlihat pada Gambar 27. Gambar 28 menyajikan kode program untuk algoritma forum diskusi.



Gbr 27. Halaman Forum Diskusi

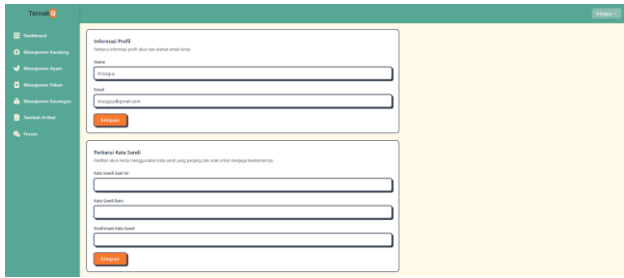
```
class TopicView extends Component
{
    public $topic;
    public function mount($topic)
    {
        try {
            $this->topic = is_numeric($topic)
                ? Topic::findOrFail($topic)
                : $topic;
            Log::info('Topic loaded', [
                'topic_id' => $this->topic->id,
                'topic_title' => $this->topic->title
            ]);
        } catch (\Exception $e) {
            Log::error('Topic loading error', [
                'error' => $e->getMessage(),
                'topic' => $topic
            ]);

            session()->flash('error', 'Topic not found');
            return redirect()->route('forum.index');
        }
    }
    public function render()
    {
        $topic = Topic::with(['user', 'comments'])
            ->findOrFail($this->topic->id);
        return view('livewire.forum.topic-view');
    }
}
```

Gbr 28. Kode Program Halaman Forum Diskusi

10)Halaman Profil

Halaman profil menampilkan informasi pribadi *user*, seperti yang terlihat pada Gambar 29. Gambar 30 menyajikan kode program untuk algoritma profil.



Gbr 29. Halaman Profil

```
public function updateProfileInformation(): void
{
    $user = Auth::user();

    $validated = $this->validate([
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'lowercase', 'email', 'max:255', Rule::unique($user->class)->ignore($user->id)],
    ], [
        'name.required' => 'Nama harus diisi.',
        'name.string' => 'Nama harus berupa teks.',
        'name.max' => 'Nama maksimal 255 karakter.',
        'email.required' => 'Email harus diisi.',
        'email.string' => 'Email harus berupa teks.',
        'email.lowercase' => 'Email harus menggunakan huruf kecil.',
        'email.email' => 'Format email tidak valid.',
        'email.max' => 'Email maksimal 255 karakter.',
        'email.unique' => 'Email sudah terdaftar di sistem.',
    ]);
}
```

Gbr 30. Kode Program Halaman Profil

D. Testing

Secara konseptual, langkah pertama dalam penilaian adalah mengubah skor mentah dari setiap item menjadi skor yang sudah disesuaikan, dengan rentang nilai dari 0 hingga 4. Penyesuaian skor berbeda antara item bernomor ganjil dan genap. Item nomor 1, 3, 5, 7, dan 9 adalah pernyataan positif. Sementara item 2, 4, 6, 8, dan 10 merupakan pernyataan negatif. Kemudian nilai setiap pernyataan dijumlahkan. Nilai ditentukan dengan menggunakan skala penilaian yang diterapkan oleh sekolah (misalnya 90-100 = A, 80-89 = B, 70-79 = C, 60-69 = D, 0-59 = F) [8]. Tabel 2 menampilkan hasil *testing* menggunakan *System Usability Testing (SUS)* yang didapat dari lima responden.

TABEL II
HASIL *SYSTEM USABILITY TESTING (SUS)*

Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total	Skor SUS
Resp-1	4	2	5	3	5	2	5	3	4	2	35	77,5
Resp-2	4	2	5	2	5	2	4	2	4	2	32	80
Resp-3	4	2	4	2	4	2	4	2	4	2	30	75
Resp-4	5	2	3	4	5	3	5	2	2	3	34	65
Resp-5	5	3	5	3	5	3	5	3	5	3	30	75

Berdasarkan hasil *System Usability Scale (SUS)*, masing-masing dari lima responden memberikan skor: 77,5; 80; 75; 65; dan 75. Jika dihitung rata-rata, skor SUS secara keseluruhan adalah 74,5. Berdasarkan klasifikasi skala penilaian yang digunakan dalam penelitian [8], skor antara 70 hingga 79 masuk dalam kategori C yang berarti cukup baik. Hasil ini menunjukkan bahwa *website* termasuk dalam kategori yang

cukup baik dalam hal kemudahan bagi *user*. Nilai tersebut juga menunjukkan bahwa sebagian besar responden mampu menggunakan *website* dengan baik.

E. Maintenance

Maintenance adalah tahap dimana *website* yang sudah dikembangkan dan diuji akan diawasi supaya *website* tetap berjalan lancar. Maintenance difokuskan pada pemantauan performa *website* serta memperbaiki *bug* yang terjadi saat *user* menggunakan *website*. Proses maintenance bertujuan supaya *website* tetap berjalan dengan baik dan dapat mendukung manajemen peternakan ayam *broiler* secara berkelanjutan.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa *website* manajemen peternakan ayam *broiler* yang telah dikembangkan memiliki fitur unggulan yang efisien dalam membantu peternak ayam *broiler* diantaranya manajemen kandang, manajemen populasi ayam berdasarkan kesehatannya, manajemen pakan, manajemen keuangan, *dashboard* visualisasi data dan analisis tren pertumbuhan atau kematian ayam, serta forum diskusi yang memungkinkan para peternak bertukar informasi sehingga mereka dapat menambah wawasan.

Secara umum, *website* yang dikembangkan telah memberi dampak positif dalam manajemen peternakan ayam *broiler* untuk meningkatkan produktivitas ayam *broiler* dan dapat memaksimalkan pendapatan peternak. Namun dalam penelitian ini hasil *System Usability Scale (SUS)* yang diperoleh dari lima responden menunjukkan bahwa *website* ini masuk ke dalam kategori cukup baik dalam hal kemudahan penggunaan, dengan skor akhir sebesar 74,5. Skor ini menunjukkan bahwa meskipun *website* sudah berjalan dengan baik dan dapat digunakan dengan mudah, masih ada beberapa hal yang bisa ditingkatkan pada penelitian berikutnya, terutama dalam desain antarmuka dan kenyamanan *user* saat berinteraksi dengan *website*. Selain itu, berdasarkan hasil penelitian dan implementasi sistem, saran pengembangan lebih lanjut yakni menambahkan fitur prediksi kemungkinan kematian ayam dengan data historis berdasarkan gejala penyakit atau perubahan lingkungan yang drastis. Berdasarkan pola ini, sistem dapat memprediksi kemungkinan kematian ayam dalam waktu dekat dan memberi peringatan dini kepada peternak untuk mengambil tindakan preventif.

V. DAFTAR PUSTAKA

- [1] D. Harsono, Atang, dan Mansuri, "Perancangan Sistem Informasi Monitoring Pertumbuhan Ayam Broiler Berbasis Web", *Prosiding Seminar Nasional Universitas Borobudur 2022*, Vol. 1, No. 1 pp. 189-209, 2022, doi: 10.37721/psdunbor.v1i1.25
- [2] T. Teguh, A. Permadi, dan N. Saurina, "Sistem Informasi Pengolahan Data Distributor Berbasis Web Pada CV. Ayam Adi Broiler", *Melek IT: Information Technology Journal*, Vol. 8, No. 1, pp. 27-36, 2022, doi: 10.30742/melekitjournal.v8i1.195.
- [3] Yulia Bagas Sri Dastiya, "Rancang Bangun Sistem Informasi Manajemen Berbasis Web di Peternakan Ayam

- Nusa Jaya”, *Diploma Thesis, Politeknik Negeri Jember*, 2022.
- [4] Martharissa Dwi Ayu Safitri, “Pengembangan Sistem Informasi Pengelolaan Ternak Ayam Petelur”, *Diploma Thesis, Politeknik Negeri Jember*, 2022.
- [5] M. Murniawati, A. Susanto, dan A. A. Riadi, “Sistem Informasi Pengelolaan Peternakan Ayam (Studi Kasus pada Peternakan Ayam Basiron Kudus)”, *JATI (Jurnal Mahasiswa Teknik Informatika)*, Vol. 8, No. 1, pp. 1200-1206, 2024, doi: 10.36040/jati.v8i1.8953.
- [6] S. Sintaro, D. Pandiangan, N. Nainggolan, A. B. Johannes, A. R. Van Gobel, dan V. P. G. Nainggolan, “Pembuatan Website Sebagai Media Informasi Digital pada Biovina Herbal”, *Journal of Social Sciences and Technology for Community Service*, Vol. 4, No. 2, pp. 285-289, 2023, doi: 10.33365/jsstcs.v4i2.3354.
- [7] A. A. Solehudin, N. Wahyu, N. Fariz, R. F. Permana, dan A. Saifudin, “Rancang Bangun Digitalisasi Persediaan Barang Berbasis Web Menggunakan Metode Waterfall”, *LOGIC: Jurnal Ilmu Komputer dan Pendidikan*, Vol 1, No. 4, pp. 1000–1005, 2023.
- [8] H. A. Rosyid, D. P. Rakhmadani, dan S. D. Alika, “Evaluasi Usability pada Aplikasi OVO Menggunakan Metode System Usability Scale (SUS)”, *JURIKOM (Jurnal Riset Komputer)*, Vol. 9, No. 6, pp. 1808-1817, 2022, doi: 10.30865/jurikom.v9i6.5073.
- [9] S. Supiyandi, M. Zen, C. Rizal, dan M. Eka, “Perancangan Sistem Informasi Desa Tomuan Holbung Menggunakan Metode Waterfall,” *JURIKOM (Jurnal Riset Komputer)*, Vol. 9, No. 2, pp. 274-283, 2022, doi: 10.30865/jurikom.v9i2.3986.
- [10] R. Candana, B. Zaman, dan S. Bahri, “Analisis Antarmuka Aplikasi Navyfriend Menggunakan Metode Usability Testing”, *KHARISMA Tech*, Vol. 19, No. 1, pp. 127-138, 2024, doi: 10.55645/kharismatech.v19i1.442
- [11] A. P. Mulia, P. R. Piri, and C. Tho, “Usability Analysis of Text Generation by Chat GPT Open AI Using System Usability Scale Method”, *Procedia Computer Science*, Vol. 227, pp. 381–388, 2023, doi: 10.1016/j.procs.2023.10.537.