

Rancang Bangun Pemetaan Peternakan Ayam di Kota Jambi Berbasis WEBGIS menggunakan Metode DevOps

Febri Dristya, Helza Triana, Muhammad Hadi Saputra*
Teknologi Rekayasa Perangkat Lunak, Politeknik Jambi

ABSTRACT

This research aims to design and develop a chicken farm mapping system in Jambi City based on WebGIS using the DevOps method. The system is designed to make it easier for users to find information about chicken farm locations geographically and to support administrators in managing farm data more efficiently. The DevOps method is applied to enhance integration and collaboration between development and operations teams, which includes the stages of planning, development, integration, testing, deployment, monitoring, and continuous improvement. In the initial phase, the system was designed using Unified Modeling Language (UML), followed by a structured database design to store essential information such as farm name, address, type of chicken, operating hours, and geographic coordinates. The system's user interface is designed to be user-friendly, making it easier for users to view farm data through an interactive map and enabling administrators to input, edit, and delete data. A monitoring system is also implemented to ensure system stability and security, using monitoring tools such as Prometheus. The research results indicate that the DevOps approach is effective in improving the quality of system development and deployment, resulting in a reliable and easily accessible WebGIS system. This system is expected to be a solution for managing chicken farm information in Jambi City and contribute to the geographic mapping of poultry farms in the region.

Article History

Received: 25-11-2024
Revised : 30-12-2024
Accepted: 30-12-2024

Keywords

WebGIS,
DevOps,
Pemetaan Peternakan,
Sistem Informasi Geografis,
Kota Jambi

ABSTRAK

Penelitian ini bertujuan untuk merancang dan mengembangkan sistem pemetaan peternakan ayam di Kota Jambi berbasis WebGIS menggunakan metode DevOps. Sistem ini dirancang untuk mempermudah pengguna dalam mencari informasi lokasi peternakan ayam secara geografis serta mendukung admin dalam mengelola data peternakan dengan lebih efisien. Metode DevOps diterapkan untuk meningkatkan integrasi dan kolaborasi antara tim pengembangan dan tim operasional, yang meliputi tahapan perencanaan, pengembangan, integrasi, pengujian, penerapan, pemantauan, dan peningkatan berkelanjutan. Pada tahap awal, sistem dirancang menggunakan Unified Modeling Language (UML), diikuti dengan desain database yang terstruktur untuk menyimpan informasi penting seperti nama peternakan, alamat, jenis ayam, jam operasional, dan koordinat geografis. Antarmuka sistem dirancang secara user-friendly untuk memudahkan pengguna dalam melihat data peternakan melalui peta interaktif serta memungkinkan admin untuk menginput, mengedit, dan menghapus data. Sistem monitoring juga diterapkan untuk menjaga stabilitas dan keamanan sistem, dengan alat pemantauan seperti Prometheus. Hasil penelitian menunjukkan bahwa pendekatan DevOps efektif dalam meningkatkan kualitas pengembangan dan penerapan sistem, dengan hasil akhir berupa sistem WebGIS yang handal dan mudah diakses. Sistem ini diharapkan dapat menjadi solusi dalam pengelolaan informasi peternakan ayam di Kota Jambi, serta memberikan kontribusi terhadap pemetaan peternakan secara geografis di wilayah tersebut.

PENDAHULUAN

Peternakan ayam merupakan salah satu sektor pertanian yang berperan penting dalam pemenuhan kebutuhan protein hewani di Indonesia. Kota Jambi, sebagai salah satu kota dengan populasi penduduk yang terus meningkat, memiliki kebutuhan yang signifikan terhadap produk unggas, khususnya daging ayam dan telur. Ketersediaan peternakan ayam yang tersebar di berbagai wilayah Kota Jambi merupakan komponen vital dalam rantai pasok makanan lokal. Namun, distribusi informasi mengenai lokasi dan kapasitas produksi peternakan ayam di kota ini belum terkelola dengan baik. Informasi yang tersedia sering kali terfragmentasi dan sulit diakses, baik oleh pemerintah daerah, masyarakat, maupun peternak itu sendiri.

Pemetaan geografis[1] peternakan ayam di Kota Jambi memiliki peran strategis dalam membantu berbagai pihak, mulai dari peternak, pemerintah, hingga konsumen. Dengan adanya peta yang akurat dan mudah diakses, pemerintah dapat merancang kebijakan yang lebih efektif dalam mendukung sektor peternakan ayam, seperti penentuan zonasi peternakan, pengawasan kesehatan hewan, serta distribusi bantuan dan fasilitas. Di sisi lain, konsumen dan pedagang dapat menggunakan peta ini untuk mengetahui lokasi peternakan terdekat, sehingga mendukung distribusi yang lebih efisien. Oleh karena itu, diperlukan sebuah sistem yang mampu memetakan peternakan ayam secara komprehensif dan mudah diakses oleh semua pihak.

Salah satu solusi yang dapat diterapkan untuk mengatasi masalah ini adalah dengan memanfaatkan teknologi WebGIS (Geographic Information System berbasis web)[2][3]. WebGIS memungkinkan pengolahan dan visualisasi data spasial melalui jaringan internet, sehingga memudahkan akses dan interaksi dengan informasi geospasial[4]. Melalui WebGIS, pengguna dapat dengan mudah mengakses data terkait peternakan ayam di Kota Jambi, seperti lokasi peternakan, luas lahan, kapasitas produksi, serta kondisi lingkungan di sekitar peternakan. Dengan demikian, WebGIS[5] dapat menjadi alat yang efektif untuk mendukung pengelolaan sektor peternakan ayam di Kota Jambi.

Selain itu, pengembangan sistem WebGIS memerlukan pendekatan pengembangan perangkat lunak yang terstruktur dan efisien[6]. Salah satu metode yang dapat digunakan adalah DevOps[7]. Metode DevOps merupakan pendekatan pengembangan perangkat lunak yang mengintegrasikan pengembangan (development) dan operasi (operations) dalam satu alur kerja yang terpadu[8][9]. DevOps menekankan kolaborasi antara tim pengembang dan tim operasional, serta penggunaan otomatisasi untuk meningkatkan efisiensi dan kualitas pengembangan perangkat lunak[10][11]. Dengan metode DevOps, proses pengembangan WebGIS dapat dilakukan secara iteratif dan responsif terhadap perubahan kebutuhan pengguna, serta meminimalkan kesalahan yang mungkin terjadi dalam pengoperasian sistem.

METODE

DevOps adalah pendekatan kolaboratif dalam pengembangan perangkat lunak yang menggabungkan praktik pengembangan (development) dan operasional (operations) untuk mempercepat siklus rilis perangkat lunak yang bias di lihat pada gambar 1[12]. Prinsip utama DevOps adalah Continuous Integration (CI) dan Continuous Deployment (CD), yang memungkinkan pengembang untuk menggabungkan kode secara rutin, menjalankan pengujian otomatis, dan menerapkan kode baru ke lingkungan produksi secara cepat[13]. Ini membantu meningkatkan kualitas perangkat lunak dan mengurangi waktu yang dibutuhkan untuk merilis fitur atau perbaikan.



Gambar 1. Siklus DevOps Services

Selain itu, DevOps menggunakan konsep Infrastructure as Code (IaC) untuk mengelola infrastruktur seperti kode yang dapat diotomatisasi, serta monitoring dan logging untuk memantau kinerja aplikasi secara real-time. Kolaborasi antara tim pengembang dan operasional juga diutamakan, memanfaatkan alat komunikasi dan dokumentasi seperti Slack dan JIRA untuk memfasilitasi alur kerja yang efisien.

Tahapan utama dalam metode DevOps biasanya mencakup langkah-langkah berikut[14][15]:

1. Planning (Perencanaan)

Deskripsi: Tahap ini melibatkan identifikasi kebutuhan pengguna, tujuan sistem, dan fitur-fitur yang akan dikembangkan. Proses ini melibatkan kolaborasi antara tim pengembang dan tim operasional untuk memastikan bahwa semua kebutuhan teknis dan operasional diperhitungkan sejak awal.

Hasil: Roadmap dan backlog yang menjelaskan fitur-fitur yang harus dikembangkan.

2. Development (Pengembangan)

Deskripsi: Pada tahap ini, kode perangkat lunak mulai ditulis dan fitur-fitur mulai dikembangkan berdasarkan rencana yang telah disusun. Pengembangan dilakukan dengan menggunakan praktik pengkodean yang efisien serta memanfaatkan sistem kontrol versi seperti Git untuk mengelola kode.

Hasil: Kode perangkat lunak dan pengembangan fitur yang dapat diuji.

3. Integration (Integrasi)

Deskripsi: Pada tahap integrasi, kode yang dikembangkan oleh berbagai tim atau pengembang individu digabungkan ke dalam satu sistem. Sistem Continuous Integration (CI) otomatisasi digunakan untuk memeriksa kualitas kode, melakukan pengujian unit, dan memastikan bahwa setiap komponen dapat berfungsi bersama.

Hasil: Kode yang telah diintegrasikan dengan sistem utama dan telah diuji secara otomatis.

4. Testing (Pengujian)

Deskripsi: Pengujian dilakukan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditetapkan. Pengujian mencakup pengujian otomatis (unit test, integration test) serta pengujian manual untuk mendeteksi bug atau error.

Hasil: Perangkat lunak yang bebas dari bug dan siap untuk di deploy.

5. Deployment (Penerapan)

Deskripsi: Tahap ini melibatkan penerapan perangkat lunak ke lingkungan produksi (deployment). Dengan Continuous Delivery (CD), perangkat lunak dapat di-deploy secara otomatis ke lingkungan produksi setelah melewati proses pengujian. Deployment sering kali dilakukan secara bertahap (staging) untuk meminimalkan risiko.

Hasil: Perangkat lunak telah diterapkan di lingkungan produksi dan siap digunakan oleh pengguna akhir.

6. Monitoring (Pemantauan)

Deskripsi: Setelah perangkat lunak di deploy, pemantauan dilakukan untuk memastikan bahwa sistem berjalan sesuai harapan dan tidak ada masalah yang muncul selama penggunaan. Metode ini menggunakan alat pemantauan seperti Prometheus atau Nagios untuk melacak kinerja, respons server, serta masalah keamanan.

Hasil: Data pemantauan yang menunjukkan kinerja aplikasi serta laporan terkait masalah atau potensi perbaikan.

7. Feedback (Umpan Balik)

Deskripsi: Setelah sistem berjalan, feedback dari pengguna dan pemantauan dikumpulkan untuk menemukan area perbaikan. Feedback ini akan digunakan dalam siklus pengembangan berikutnya untuk iterasi yang lebih baik dan peningkatan berkelanjutan.

Hasil: Umpan balik yang digunakan untuk menyempurnakan sistem dalam siklus berikutnya.

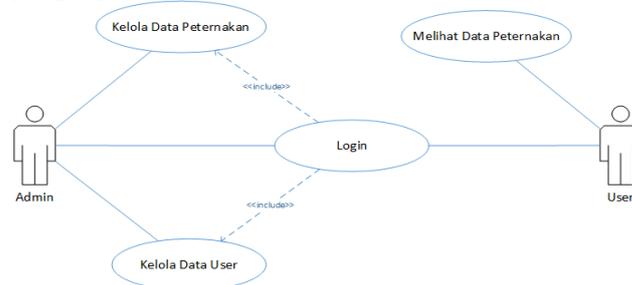
8. Continuous Improvement (Peningkatan Berkelanjutan)

Deskripsi: Dalam metode DevOps, siklus pengembangan tidak berhenti setelah deployment. Proses ini berlanjut secara berkesinambungan dengan penerapan perbaikan-perbaikan berdasarkan hasil monitoring dan feedback, serta pengembangan fitur baru untuk menyesuaikan dengan kebutuhan pengguna.

Hasil: Sistem yang selalu diperbarui dan dioptimalkan seiring waktu

HASIL DAN PEMBAHASAN

Dalam tahapan pertama yaitu *planning* atau perencanaan disini kita rancang sistem nya sesuai dengan kebutuhan user menggunakan *UML*. Rancang Bangun[16][17][18] sistem yang pertama kali yaitu use case diagram yang dijelaskan dalam Gambar 2



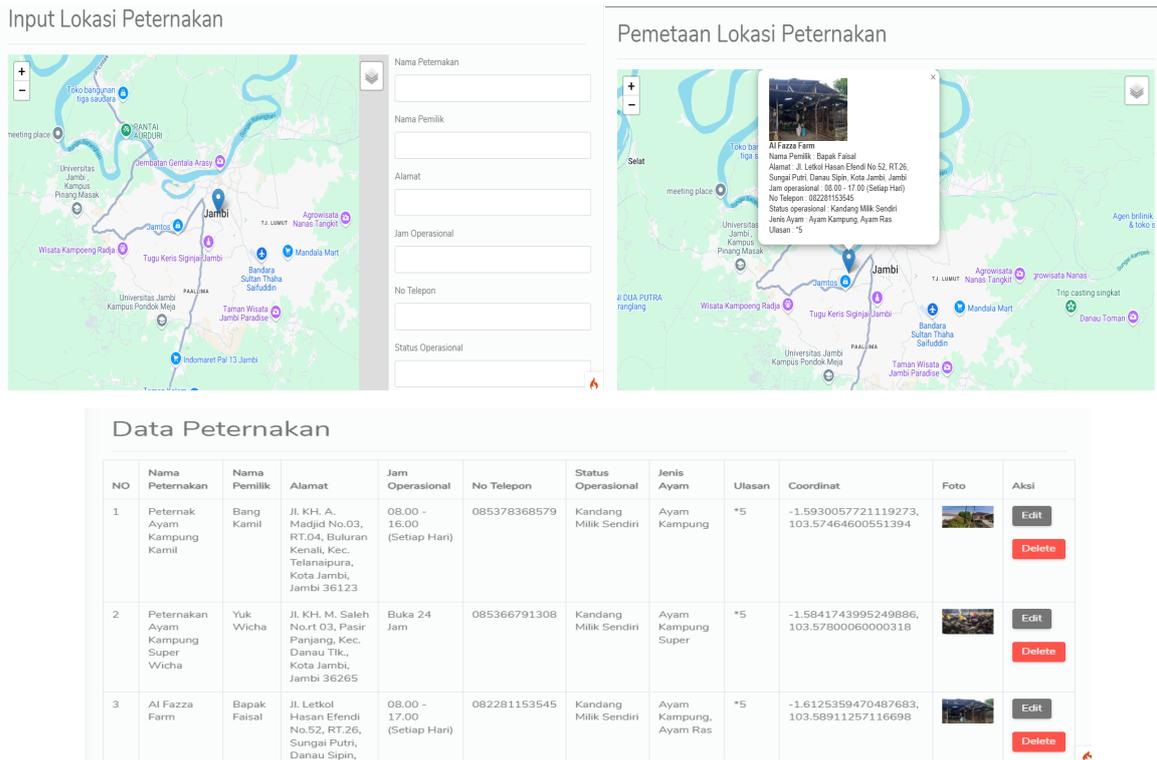
Gambar 2. Use Case Diagram

Pada use case ini admin berperan sebagai pengelola data peternakan dan data user, sedangkan user hanya bisa melihat data peternakan yang ada. Selanjutnya desain database yang dijelaskan pada Gambar 3 dan field yang dibutuhkan adalah : nama_peternakan, nama_pemilik, alamat, jam_operasional, no_telepon, status_operasional, jenis_ayam, ulasan, latitude, longitude, foto,peternakan.

#	Nama	Jenis
1	id_peternakan	int(11)
2	nama_peternakan	varchar(50)
3	nama_pemilik	varchar(50)
4	alamat	text
5	jam_operasional	text
6	no_telepon	varchar(50)
7	status_operasional	text
8	jenis_ayam	varchar(50)
9	ulasan	text
10	latitude	varchar(255)
11	longitude	varchar(255)
12	foto_peternakan	varchar(255)

Gambar 3. Desain Database

Setelah rancangan database tercipta selanjutnya rancangan antar muka yang bisa dilihat pada Gambar 4 dari aplikasi seperti : input lokasi peternakan, pemetaan lokasi peternakan, list lokasi peternakan. Dalam antar muka input lokasi peternakan di bagian kiri terdapat map untuk mencari posisi dari peternakan tersebut, dan di sebelah kanan terdapat form input yang terdiri dari nama_peternakan, nama_pemilik, alamat, jam_operasional, no_telepon, status_operasional, jenis_ayam, ulasan, latitude, longitude, foto, dan peternakan. Pada antar muka untuk pemetaan lokasi peternakan ditampilkan map yang berisi marker seluruh data peternakan ayam yang berada di seluruh kota jambi, untuk menu data peternakan berisi seluruh data dan ada dua tombol untuk edit dan delete data yang bias dilakukan oleh admin. Dalam tahapan kedua *development* atau pengembangan, kode perangkat lunak mulai ditulis dan fitur-fitur mulai dikembangkan berdasarkan rencana yang telah disusun, pada gambar 5 bisa dilihat contoh salah satu coding untuk input lokasi pemetaan.



Gambar 4. Antar Muka Sistem

```

<div class="row">
  <div class="col-sm-8">
    <div id="map" style="width: 100%; height: 100vh;"></div>
  </div>
  <div class="col-sm-4">
    <div class="row">
      <div class="form-group">
        <input type="text" name="nama_peternakan" />
        <p class="text-danger"><? = isset($errors['nama_peternakan']) == isset($errors['nama_peternakan']) ? validation_show_error('nama_peternakan') : '' ?></p>
      </div>
      <div class="form-group">
        <input type="text" name="nama_pemilik" />
        <p class="text-danger"><? = isset($errors['nama_pemilik']) == isset($errors['nama_pemilik']) ? validation_show_error('nama_pemilik') : '' ?></p>
      </div>
      <div class="form-group">
        <input type="text" name="alamat" />
        <p class="text-danger"><? = isset($errors['alamat']) == isset($errors['alamat']) ? validation_show_error('alamat') : '' ?></p>
      </div>
      <div class="form-group">
        <input type="text" name="jam_operasional" />
        <p class="text-danger"><? = isset($errors['jam_operasional']) == isset($errors['jam_operasional']) ? validation_show_error('jam_operasional') : '' ?></p>
      </div>
      <div class="form-group">
        <input type="text" name="no_telepon" />
        <p class="text-danger"><? = isset($errors['no_telepon']) == isset($errors['no_telepon']) ? validation_show_error('no_telepon') : '' ?></p>
      </div>
      <div class="form-group">
        <input type="text" name="status_operasional" />
        <p class="text-danger"><? = isset($errors['status_operasional']) == isset($errors['status_operasional']) ? validation_show_error('status_operasional') : '' ?></p>
      </div>
    </div>
    <div class="form-group">
      <input type="text" name="jenis_ayam" />
      <p class="text-danger"><? = isset($errors['jenis_ayam']) == isset($errors['jenis_ayam']) ? validation_show_error('jenis_ayam') : '' ?></p>
    </div>
    <div class="form-group">
      <input type="text" name="ulasan" />
      <p class="text-danger"><? = isset($errors['ulasan']) == isset($errors['ulasan']) ? validation_show_error('ulasan') : '' ?></p>
    </div>
    <div class="form-group">
      <input type="text" name="latitude" id="Latitude" />
      <p class="text-danger"><? = isset($errors['latitude']) == isset($errors['latitude']) ? validation_show_error('latitude') : '' ?></p>
    </div>
    <div class="form-group">
      <input type="text" name="longitude" id="Longitude" />
      <p class="text-danger"><? = isset($errors['longitude']) == isset($errors['longitude']) ? validation_show_error('longitude') : '' ?></p>
    </div>
    <div class="form-group">
      <input type="file" name="foto_peternakan" accept="image/*" />
      <p class="text-danger"><? = isset($errors['foto_peternakan']) == isset($errors['foto_peternakan']) ? validation_show_error('foto_peternakan') : '' ?></p>
    </div>
    <div class="form-group">
      <input type="submit" value="Simpan" />
    </div>
  </div>
</div>
</pre>


```

<div class="form-group">
 <input type="text" name="nama_peternakan" />
 <p class="text-danger"><? = isset($errors['nama_peternakan']) == isset($errors['nama_peternakan']) ? validation_show_error('nama_peternakan') : '' ?></p>
</div>
<div class="form-group">
 <input type="text" name="nama_pemilik" />
 <p class="text-danger"><? = isset($errors['nama_pemilik']) == isset($errors['nama_pemilik']) ? validation_show_error('nama_pemilik') : '' ?></p>
</div>
<div class="form-group">
 <input type="text" name="alamat" />
 <p class="text-danger"><? = isset($errors['alamat']) == isset($errors['alamat']) ? validation_show_error('alamat') : '' ?></p>
</div>
<div class="form-group">
 <input type="text" name="jam_operasional" />
 <p class="text-danger"><? = isset($errors['jam_operasional']) == isset($errors['jam_operasional']) ? validation_show_error('jam_operasional') : '' ?></p>
</div>
<div class="form-group">
 <input type="text" name="no_telepon" />
 <p class="text-danger"><? = isset($errors['no_telepon']) == isset($errors['no_telepon']) ? validation_show_error('no_telepon') : '' ?></p>
</div>
<div class="form-group">
 <input type="text" name="status_operasional" />
 <p class="text-danger"><? = isset($errors['status_operasional']) == isset($errors['status_operasional']) ? validation_show_error('status_operasional') : '' ?></p>
</div>
</pre>
<script>
 var peta1 = L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token=pk.eyJ1IjoibW9uZmF0b3Q', {
 attribution: 'Map data © OpenStreetMap contributors, ' +
 'CC-BY-SA',
 'imagery': 'mapbox://www.mapbox.com/mapbox-com',
 id: 'mapbox/streets-v11'
 });
 const map = L.map('map', {
 center: [-1.61032688713177, 103.61433850068897],
 zoom: 11,
 layers: [peta2]
 });
 const baseLayers = [
 {
 'default': peta4,
 'Gmaps': peta2,
 'OSM-Hybrid': peta3,
 'OSM-Mapbox': peta1,
 'OSM': peta5,
 'Dark-OSM': peta6,
 'Carto-OSM': peta7,
 'ArcGIS': peta8,
 'Gmaps-Marker': peta9,
 'Gmaps': peta10,
 }
];
 const layerControl = L.control.layers(baseLayers).addTo(map);
 //get koordinat
 var latInput = document.querySelector('[name=latitude]');
 var lngInput = document.querySelector('[name=longitude]');
 var posisi = document.querySelector('[name=posisi]');
 var curLocation = [-1.61032688713177, 103.61433850068897];
</script>

```


```

Gambar 5. Kode Program untuk Input Lokasi Pemetaan

Dalam tahapan selanjutnya integrasi, kode yang dikembangkan oleh berbagai tim atau pengembang individu digabungkan ke dalam satu sistem. Sistem Continuous Integration (CI) otomatisasi digunakan untuk memeriksa kualitas kode, melakukan pengujian unit, dan memastikan

bahwa setiap komponen dapat berfungsi bersama. Pada sistem Pemetaan Peternakan Ayam di Kota Jambi Berbasis WEBGIS ini semua kode berfungsi sebagaimana kebutuhannya.

Setelah itu di lakukan pengujian perangkat lunak[19] nya untuk memastikan kalau setiap kode yang di ciptakan bebas dari bug dan siap untuk dilakukan ketahapan berikutnya yaitu deployment (penerapan). Tahap ini melibatkan penerapan perangkat lunak ke lingkungan produksi (deployment). Dengan Continuous Delivery (CD), perangkat lunak dapat dideploy secara otomatis ke lingkungan produksi setelah melewati proses pengujian.

Setelah perangkat lunak di deploy, pemantauan dilakukan untuk memastikan bahwa sistem berjalan sesuai harapan dan tidak ada masalah yang muncul selama penggunaan. Metode ini menggunakan alat pemantauan seperti Prometheus atau Nagios untuk melacak kinerja, respons server, serta masalah keamanan. Setelah sistem berjalan, feedback dari pengguna dan pemantauan dikumpulkan untuk menemukan area perbaikan

Dalam metode DevOps, siklus pengembangan tidak berhenti setelah deployment. Proses ini berlanjut secara berkesinambungan dengan penerapan perbaikan-perbaikan berdasarkan hasil monitoring dan feedback, serta pengembangan fitur baru untuk menyesuaikan dengan kebutuhan pengguna.

KESIMPULAN

Pengembangan Sistem Pemetaan Berbasis WebGIS berhasil merancang dan membangun sistem pemetaan peternakan ayam berbasis WebGIS yang dapat diakses secara online oleh pengguna dan admin. Sistem ini memungkinkan visualisasi lokasi peternakan ayam di Kota Jambi, yang sangat membantu pengguna dalam mencari informasi peternakan ayam, serta memudahkan admin dalam mengelola data peternakan. Metode DevOps yang diterapkan dalam pengembangan sistem terbukti meningkatkan efisiensi proses pengembangan dan penerapan sistem. Dengan adanya integrasi antara tim pengembangan dan operasional melalui tahapan perencanaan, pengembangan, integrasi, pengujian, penerapan, serta pemantauan yang kontinu, sistem dapat dirancang dengan lebih baik, diuji dengan cermat, dan diterapkan dengan minim risiko. Hal ini juga memungkinkan pembaruan berkelanjutan berdasarkan feedback pengguna.

Desain Database yang Terstruktur dengan berbagai atribut yang relevan, seperti nama peternakan, alamat, jenis ayam, jam operasional, lokasi geografis, dan ulasan. Desain ini memungkinkan sistem untuk menyimpan data peternakan dengan efisien, sehingga informasi dapat diakses dan diperbarui dengan mudah oleh admin. Antar muka sistem yang dirancang memungkinkan pengguna untuk melihat data peternakan ayam dalam format peta dengan marker yang interaktif, serta memungkinkan admin menginput, mengedit, dan menghapus data peternakan. Hal ini memberikan pengalaman pengguna yang mudah dan nyaman, baik untuk admin maupun pengguna umum. Dengan penggunaan alat monitoring seperti Prometheus atau Nagios, sistem dapat dipantau kinerjanya secara terus-menerus untuk memastikan stabilitas dan keamanan. Pemantauan ini memungkinkan identifikasi masalah lebih awal dan mendukung penerapan pembaruan secara berkelanjutan sesuai kebutuhan.

DAFTAR PUSTAKA

- [1] F. Dristyan, M. Mardalius, Y. A. M, and M. P. Ningrum, "Pemetaan Desa-desa di Kabupaten Labuhan Batu Utara Menggunakan Leaflet API Berbasis WEB," *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 13, no. 2, p. 148, 2023, doi: 10.36448/expert.v13i2.3340.
- [2] Mardalius, F. Dristyan, R. Aulia, ega evinda Putri, and I. Syafrinal, "Geographic information system for Covid-19 vaccine distribution with Laravel framework," *AIP Conf. Proc.*, 2024.
- [3] N. Pratiwi, E. Kurniawan, and F. Dristyan, "Pemetaan Lahan Karet Dan Kelapa Sawit Pada PT. BSP Tbk Kab. Asahan Berbasis WebGis," *J. Comput. Sci. Technol.*, vol. 1, no. 3, pp. 108–117, 2023, [Online]. Available: <https://jurnal.padangtekno.com/index.php/jocstec/article/view/169>
- [4] Mardalius and F. Dristyan, "Pemanfaatan Library Leaflet Pada GIS Sekolah Di Dinas Pendidikan Kabupaten Asahan Menggunakan Framework Codeigniter 4," *J. Teknol. Sist.*

- Inf. dan Sist. Komput. TGD*, vol. 6, no. 1, pp. 157–163, 2023, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jsk/index>
- [5] M. Mardalius, F. Dristyan, and A. Syafnur, “Sistem Informasi Geografis Penyebaran Covid-19 Di Kabupaten Asahan Menggunakan Framework Codeigniter 4,” *J. Sci. Soc. Res.*, vol. 4, no. 3, pp. 347–351, 2021, [Online]. Available: <http://jurnal.goretanpena.com/index.php/JSSR>
- [6] N. L. Utami, A. Nazir, E. Budianita, and F. Insani, “Perancangan Antarmuka Aplikasi foodcare menggunakan metode user centered design,” *J. Comput. Sci. Inf. Technol. (CoSciTech)*, vol. 5, no. 1, pp. 75–83, 2024.
- [7] M. A. Akbar, S. Rafi, A. A. Alsanad, S. F. Qadri, A. Alsanad, and A. Alothaim, “Toward Successful DevOps: A Decision-Making Framework,” *IEEE Access*, vol. 10, pp. 51343–51362, 2022, doi: 10.1109/ACCESS.2022.3174094.
- [8] A. Bahaa, Y. Mostafa, and - Mahmoud, “Enhancing Lean Software Development by using Devops Practices,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 7, pp. 267–277, 2017, doi: 10.14569/ijacsa.2017.080736.
- [9] R. T. Yarlagadda, “How DevOps Enhances the Software Développement Quality,” *Int. J. Creat. Res. Thoughts*, vol. 7, no. 3, pp. 358–364, 2019.
- [10] N. M. Noorani, A. T. Zamani, M. Alenezi, M. Shameem, and P. Singh, “Factor Prioritization for Effectively Implementing DevOps in Software Development Organizations: A SWOT-AHP Approach,” *Axioms*, vol. 11, no. 10, pp. 1–29, 2022, doi: 10.3390/axioms11100498.
- [11] D. S. Battina, “Devops, A New Approach To Cloud Development & Testing (August 8, 2020),” *Int. J. Emerg. Technol. Innov. Res.*, vol. 7, no. 8, pp. 982–985, 2020, [Online]. Available: <https://ssrn.com/abstract=4004330>
- [12] A. Hermawan and L. P. Manik, “The Effect of DevOps Implementation on Teamwork Quality in Software Development,” *J. Inf. Syst. Eng. Bus. Intell.*, vol. 7, no. 1, p. 84, 2021, doi: 10.20473/jisebi.7.1.84-90.
- [13] D. B. Sindhu, “The challenges and Mitigation Strategies of Using DevOps during Software Development,” *Int. J. Creat. Res. Thoughts*, vol. 9, no. 1, 2021.
- [14] G. B. Ghantous and A. Q. Gill, “DevOps: Concepts, practices, tools, benefits and challenges,” *Proc. of 21st Pacific Asia Conf. Inf. Syst. “Societal Transform. Through IS/IT”*, PACIS 2017, 2017.
- [15] L. Zhu, L. Bass, and G. Champlin-Scharff, “DevOps and Its Practices,” *IEEE Softw.*, vol. 33, no. 3, pp. 32–34, 2016, doi: 10.1109/MS.2016.81.
- [16] I. Taufik, P. Benny Herlandy, and M. Novalia, “Rancang Bangun Sistem Penjadwalan Pembelajaran Menggunakan Metode Algoritma Genetika Di Smk Multi Mekanik Masmur,” *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 5, no. 1, pp. 281–290, 2024, doi: 10.37859/coscitech.v5i1.6887.
- [17] F. Dristyan, K. Priyanto, and S. Andriyani, “Rancang Bangun Dan Implementasi Siades Pada Desa Perjuangan Kab. Batu Bara,” *J. Sci. Soc. Res.*, vol. 4, no. 2, pp. 180–184, 2021, [Online]. Available: <http://jurnal.goretanpena.com/index.php/JSSR>
- [18] M. Christina, M. S. Malawat, and F. Dristyan, “Rancang Bangun Sistem Pakar Diagnosa Penyakit Tanaman Kelapa Menggunakan Metode Backward Chaining,” *J. Tek.*, vol. 1, no. 1, p. 19, 2021, doi: 10.54314/teknisi.v1i1.478.
- [19] F. Dristyan and M. Meri, “The Testing Of Library Application By Using Boundary Value Analysis,” in *Proceeding International Conference on Social, Sciences and Information Technology*, 2020, pp. 151–156. doi: 10.33330/icossit.v1i1.785.