

PENGEMBANGAN APLIKASI MANAJEMEN TAMBAK IKAN KONSUMSI BERBASIS WEB DAN MOBILE FISHEERLOG

**Rio Ferddinansya¹, Muhammad Naufal Sutardi², Rheynesta Hannover³, Fauzi Ikhsan Suswanto⁴,
Muhammad Rafi Alexander Prayoga⁵, Aditya Wicaksono⁶, Gema Parasti Mindara⁷, Inna Novianty⁸,
Lathifunnisa Fathonah⁹, Endang Purnama Giri¹⁰**

¹⁻⁶Program Studi Teknologi Rekayasa Perangkat Lunak, Sekolah Vokasi, IPB University

⁷⁻⁹Program Studi Teknologi Rekayasa Komputer, Sekolah Vokasi, IPB University

¹⁰Program Studi Ilmu Komputer, Sekolah Sains Data, Matematika, dan Informatika, IPB University

¹⁻⁹Jl. Kumbang No.14, Kampus IPB Cilibende, Kota Bogor, Jawa Barat, Indonesia

¹⁰Jl. Raya Dramaga Kampus IPB Dramaga, Kab. Bogor, Jawa Barat, Indonesia

E-mail: rioferddinansya@apps.ipb.ac.id¹, naufalsutardi@apps.ipb.ac.id², rheynestahannover@apps.ipb.ac.id³,
fauziikhsansuswanto@apps.ipb.ac.id⁴, alexanderafi@apps.ipb.ac.id⁵, adityawicaksono@apps.ipb.ac.id⁶,
gemaparasti@apps.ipb.ac.id⁷, innanovianty@apps.ipb.ac.id⁸, lathifunnisafathonah@apps.ipb.ac.id⁹,
endang_pg@apps.ipb.ac.id¹⁰

Abstrak – Manajemen pakan ikan, stok ikan, serta pemantauan karyawan adalah parameter krusial dalam kesuksesan pengelolaan tambak ikan. Dalam skala besar, pengelolaan tambak ikan akan cenderung menjadi lebih rumit dan kompleks. Memastikan setiap sumber daya manusia yang mengelola tambak ikan teredukasi dan memahami metode teknis dalam membudi daya ikan menjadi langkah krusial yang dapat dilakukan pengelola tambak ikan. Dengan aplikasi ini, manajer tambak ikan dapat dengan mudah mengelola logistik ikan mulai dari pakan dan stok, hingga monitoring perawatan tambak ikan oleh karyawan. Sarana pembelajaran melalui aplikasi akan menjadi solusi praktis sekaligus berdampak besar bagi karyawan untuk lebih memahami praktik pengelolaan tambak ikan. Aplikasi ini dikembangkan menggunakan metode *Agile* yang memudahkan pengembangan dan fleksibel sehingga memungkinkan pengembangan berkelanjutan. Website *Fisheerlog* dibangun menggunakan *framework Laravel 11*, sedangkan aplikasi mobile dikembangkan menggunakan *Java*. Aplikasi ini diharapkan bisa mempermudah pengelolaan tambak ikan baik dari segi logistik maupun sumber daya manusia yang tersedia. Selain itu, aplikasi ini diharapkan dapat mendorong minat masyarakat untuk memulai bisnis budidaya ikan konsumsi.

Kata Kunci: Logistik, Manajemen, Tambak, Pakan, Stok

I. PENDAHULUAN

Indonesia memiliki potensi perikanan budidaya yang sangat besar, dengan luas lahan mencapai 17,91 juta hektar, yang terdiri dari lahan budidaya air tawar sebesar 2,8 juta hektar (15,8%), lahan budidaya air payau 2,96 juta hektar (16,5%), dan lahan budidaya laut sebesar 12,12 juta hektar (67,7%) (DJPB, 2019). Namun, dari total potensi lahan budidaya air payau, pemanfaatannya masih terbatas pada sekitar 9,6% atau seluas 277.397 hektar. Hal ini menunjukkan adanya peluang besar yang belum optimal dimanfaatkan untuk mendukung pengembangan sektor perikanan di Indonesia.

Sesuai dengan Dokumen Rencana Strategis (Renstra) Deputy Bidang Koordinasi Sumber Daya Maritim untuk periode 2020-2024, pemerintah menargetkan peningkatan konsumsi ikan menjadi 58,3 kg/kapita/tahun pada tahun 2020 dan 60,9 kg/kapita/tahun pada tahun 2024. Namun, berdasarkan Laporan Kinerja Kementerian Kelautan dan Perikanan (KKP) tahun 2023, angka konsumsi ikan yang tercapai hanya sebesar 54,56 kg/kapita pada tahun 2020 dan 57,61 kg/kapita pada tahun

2023, yang berarti target konsumsi ikan belum terpenuhi.

Kegagalan pencapaian target ini dipengaruhi oleh berbagai isu strategis dalam pengelolaan sumber daya perikanan. Berdasarkan Laporan Kinerja Direktorat Pengelolaan Sumber Daya Ikan pada Triwulan II Tahun 2024, salah satu masalah utama adalah belum optimalnya akurasi data stok sumber daya ikan, hasil tangkapan, dan tingkat eksploitasi sumber daya ikan. Selain itu, rendahnya kepatuhan pelaku usaha dalam melaporkan hasil tangkapan ikan secara akurat, objektif, dan tepat waktu juga menjadi hambatan signifikan dalam perencanaan dan pengelolaan sektor perikanan.

Di sektor budidaya, tantangan lain yang muncul adalah rendahnya pengetahuan teknologi dan manajemen pascapanen di kalangan petani tambak. Penelitian yang dilakukan oleh Ryan & Parikesit (2024) di Desa Prasung, Kecamatan Buduran, Kabupaten Sidoarjo, tahun 2024 menunjukkan bahwa meskipun petani tambak ikan nila memiliki pengalaman cukup untuk memproduksi ikan dalam jumlah besar, mereka menghadapi keterbatasan informasi tentang

pengelolaan pascapanen yang efektif. Kondisi serupa terjadi pada petani tambak ikan bandeng di Kabupaten Pangkajene dan Kepulauan, Sulawesi Selatan, selama masa pandemi, di mana rendahnya pendidikan dan keterampilan petani dalam pemeliharaan serta peningkatan produksi ikan menjadi kendala utama (Nida, 2022).

Permasalahan ini menunjukkan bahwa potensi besar sektor perikanan budidaya di Indonesia belum sepenuhnya dimanfaatkan. Pengelolaan pascapanen yang kurang efektif, rendahnya pengetahuan teknologi, serta keterbatasan dalam pendidikan dan pelatihan petani tambak menjadi tantangan yang memerlukan perhatian serius. Oleh karena itu, penelitian ini penting untuk mengidentifikasi solusi strategis guna meningkatkan produktivitas sektor perikanan budidaya, mendukung peningkatan konsumsi ikan nasional, dan memaksimalkan potensi lahan perikanan yang ada.

II. TINJAUAN PUSTAKA

Tambak merupakan alternatif pemanfaatan sumberdaya lahan di pesisir yang hanya dapat dilakukan di air payau. Hal tersebut dikarenakan ikan yang akan dibudidayakan memerlukan air payau sebagai sarana hidup (Desi, 2019). Air payau adalah air yang salinitasnya lebih rendah dari pada salinitas rata-rata air laut normal (<35 ppm) dan lebih tinggi dari pada 0,5 ppm yang terjadi karena pencampuran antara air laut dengan air tawar baik secara alamiah maupun buatan (Leila et. al., 2021).

Pada dasarnya, setiap tambak ikan memiliki banyak kolam untuk menampung ikan-ikan. Pembagian kolam ini mempermudah pengamatan, pencatatan, serta pendataan dari ikan-ikan yang ada. Dari setiap kolam yang ada, terdapat ikan-ikan dengan kebutuhan masing-masing khususnya kebutuhan pakan. Pemberian pakan pada suatu tambak ikan salah satu hal krusial yang perlu di kelola. Hal ini karena pertumbuhan ikan yang baik dinilai dari pemberian pakannya yang disiplin, tepat waktu, dan efektif. Manajemen pemberian pakan bertujuan untuk menyesuaikan kebutuhan nutrisi ikan, kualitas pakan dan waktu yang tepat untuk memberi makan ikan. Waktu pemberian pakan yang tepat pada budidaya ikan dapat menjadikan keberhasilan dalam suatu usaha budidaya, dan apabila terjadi penggunaan pakan yang tidak efisien/berlebihan dan pemberian pakannya di waktu yang berbeda dapat menyebabkan kerugian dalam budidaya (Layla & Regitu, 2024). Semakin besar ukuran ikan semakin sedikit jumlah pakan yang diberikan, dan sebaliknya. Hal ini dikarenakan ikan berukuran kecil membutuhkan jumlah pakan yang lebih banyak karena ikan berukuran kecil masa pertumbuhannya lebih besar dari pada ikan yang berukuran besar. *Feeding habits* atau kebiasaan cara makan ikan merupakan hal yang harus diperhatikan dalam pemberian pakan suatu budidaya ikan

(Aminullah, 2024). Selain itu, untuk mengoptimalkan pakan untuk dicerna, pemberian pakan harus memperhatikan nafsu makan dan tingkat kenyangannya (Layla & Regitu, 2024).

Untuk mengedukasi karyawan tentang pengelolaan tambak, kolam, hingga pakan, modul menjadi solusi penulis dalam pengembangan aplikasi. "Modul merupakan bahan ajar mandiri yang bisa membantu peserta didik dalam mengembangkan kompetensi yang dimiliki secara mandiri sesuai dengan kemampuan yang dimiliki" (Tasya et al., 2023). Modul dinilai sebagai bahan ajar yang praktis dengan karakteristik self instructional (instruksi yang jelas), self contained (materi pembelajara yang dapat dipelajari secara mandiri), stand alone (tidak tergantung pembelajaran beradaptasi lain), dengan adaptif menyesuaikan media (mudah dengan kemajuan teknologi), dan user friendly (mudah dipahami) (Kosasih, 2021). Modul pembelajaran dapat memberi dampak positif yang berguna bagi edukasi karyawan demi meningkatkan SDM yang tersedia, mempertahankan kualitas serta meningkatkan kepercayaan kepada karyawan. Fitur modul ini akan tersedia melalui perangkat lunak *mobile* yang akan digunakan oleh karyawan dari perangkat lunak yang penulis kembangkan

III. METODE PENELITIAN

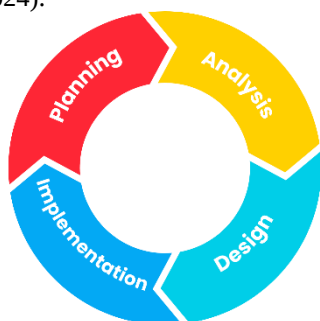
Perancangan dimulai dari penetapan timeline untuk menentukan dan mengkalkulasi durasi pengerjaan menggunakan *Jira* sebagai platform manajemen timeline. Waktu pengerjaan proyek ini dimulai dari tanggal 12 Agustus 2024 hingga 17 November 2024 atau sekitar 14 minggu.

Setiap minggunya difokuskan untuk mengerjakan bagian tertentu guna memaksimalkan potensi pengembangan.

1. Minggu 1: Pembentukan kelompok dan penentuan bagian pekerjaan menyesuaikan dengan bidang yang diperlukan dalam proses pengembangan.
2. Minggu 2: Persiapan pengembangan berisikan kegiatan seperti brainstorming, mencari latar belakang masalah yang harus diberikan solusi, hingga menentukan solusi yang dapat membantu permasalahan yang ada.
3. Minggu 3: Mendesain infrastruktur, *ERD*, *DFD*, *UML*, *API*, *UI/UX*, dan hal lain yang menjadi penentu sekaligus acuan dalam proses pengembangan perangkat lunak.
4. Minggu 4: Hasil kemajuan yang sudah dilakukan minggu-minggu sebelumnya dinilai dan diberi masukan oleh dosen pengampu agar pengerjaan proyek lebih jelas dan terarah.
5. Minggu 5 dan 6: Desain yang sudah direvisi dieksekusi melalui pengembangan front-end serta memaksimalkan potensi *User Journey* demi memberikan pengalaman terbaik terhadap pengguna.

6. Minggu 7 dan 8: Pengembangan back-end dari perangkat lunak mulai dieksekusi. Desain yang dirancang pada minggu sebelumnya dijadikan sebagai acuan dalam proses pengembangan ini.
7. Minggu 9 dan 10: Kemajuan dari pengembangan dipresentasikan kepada dosen pengampu untuk diberi masukan dan direvisi. Progres pengembangan masih terus dilakukan hingga tahap final.
8. Minggu 11: Setelah tahap pengembangan mencapai akhir, setiap fungsi dan fitur diuji coba hingga mendapat hasil memuaskan.
9. Minggu 12: Perangkat lunak dihosting agar dapat diakses secara terbuka melalui internet.
10. Minggu 13: Saat ini perangkat lunak sudah mencapai tahap akhir dan siap untuk diluncurkan.
11. Minggu 14: Perancangan perangkat lunak mencapai tahap akhir dan dipresentasikan kepada semua dosen pengampu.

Metode yang digunakan dalam perancangan perangkat lunak adalah *SDLC*. *Software Development Life Cycle (SDLC)* adalah proses mengembangkan atau mengubah suatu sistem dengan kerangka kerja yang terstruktur dan teratur dalam pengembangan perangkat lunak, serta dapat membantu mengatur proses secara sistematis (Dewa & Azizah, 2024).



Gambar 1. Ilustrasi Tahapan Metode *SDLC*

Secara umum, metode *SDLC* dibagi dalam beberapa tahap. Berikut adalah tahapan-tahapan *SDLC* menurut Dewa & Azizah (2024):

1. *Planning* (Perencanaan): *Planning* atau perencanaan memiliki peran utama dalam kerangka pengembangan proyek perangkat lunak. Tahap ini mengidentifikasi permasalahan yang ada, kemudian akan dilakukan analisis untuk menghasilkan suatu temuan.
2. *Analysis* (Analisis): Penekanan pada analisis pengembangan perangkat lunak yang akan diterapkan memerlukan pengumpulan data yang relevan. Metode pengumpulan data yang dilakukan adalah melalui riset mendalam pada jurnal dan hasil penelitian terdahulu, lalu mengidentifikasi permasalahan yang diangkat.
3. *Design* (Perancangan): Desain ini digunakan untuk menentukan bagaimana perangkat lunak

berjalan. Desain yang terstruktur dan jelas akan menjadi parameter keberhasilan perangkat lunak.

4. *Implementation* (Implementasi): Setelah tahap sebelumnya dirancang dengan baik, maka akan diimplementasikan dalam bentuk nyata melalui pengembangan web dan aplikasi.

Metode *SDLC* memiliki beberapa model, dan model yang digunakan dalam pengembangan perangkat lunak ini adalah model *Agile*. *Agile Software Development* atau sering disebut sebagai metode *Agile* adalah metodologi pengembangan perangkat lunak yang didasarkan pada proses pengerjaan yang dilakukan berulang, di mana aturan dan solusi yang disepakati dilakukan dengan kolaborasi antar tiap tim secara terorganisir dan terstruktur (Adani, 2023).



Gambar 2. Ilustrasi Tahapan Model *Agile*

Pada prosesnya, model *Agile* menerapkan metode *SDLC* yang umum digunakan namun ada beberapa tahap yang membedakannya (Adani, 2023). Berikut tahap-tahap metode *Agile* menurut Paksi et al., (2023):

1. Perencanaan (*Planning*): Pengembang dan klien mendefinisikan kebutuhan apa saja yang diperlukan dalam menentukan perangkat lunak yang akan dibangun.
2. Perancangan (*Design*): Tahap ini dapat menggunakan diagram alir atau diagram lainnya untuk menunjukkan cara kerja fitur dan menampilkan bagaimana penerapan *UI/UX* pada perangkat lunak.
3. Pengembangan (*Development*): *Programmer* melakukan pengkodean (*coding*) perangkat lunak. Pada tahap pengembangan ini, perangkat lunak akan menjalani beberapa tahap penyempurnaan agar mencakup fungsionalitas yang dibutuhkan.
4. Pengujian (*Testing*): *Bug* yang ditemukan dari tahap ini akan segera diperbaiki agar kualitas perangkat lunak terjaga.
5. Penyebaran (*Deployment*): Aplikasi web dan aplikasi mobile akan diluncurkan. Website akan dihosting dan aplikasi mobile akan tersedia di layanan distribusi digital.
6. Peninjauan (*Review*): Pengembang perangkat lunak menerima umpan balik dan masukan tentang produk lalu bekerja berdasarkan umpan balik yang telah diberikan.

Adapun kekurangan dari model *Agile* (Paksi et al., 2023): 1) Perencanaan yang sedikit sehingga

rancangan kurang terstruktur dengan jelas; 2) Menggunakan banyak tenaga serta waktu karena perubahan bisa terjadi kapan saja. Namun, terdapat beberapa kelebihan dari model *Agile*, seperti: 1) Waktu yang relatif singkat dan tidak memerlukan sumber daya yang banyak; 2) Memprioritaskan kebutuhan dan kepuasan klien, sehingga bersifat fleksibel; 3) Pengembangan memiliki fokus yang bertahap dan terprediksi, sehingga risiko dapat diminimalisir.

Adapun *framework* yang digunakan dalam proses pengembangan perangkat lunak adalah *Laravel* 11 dengan bahasa pemrograman *PHP* 8.2.25. *Laravel*, yang berbasis pada bahasa pemrograman *PHP*, telah menjadi salah satu pilihan utama bagi pengembang karena menawarkan berbagai keunggulan dalam hal kemudahan penggunaan, keamanan, dan efisiensi (Sinlae et al., 2024). *Laravel* menjadi salah satu *framework* paling populer saat ini karena kemudahan dan skalabilitasnya. Menurut Sinlae et al., ada beberapa keunggulan yang dimiliki *Laravel* sehingga menjadi *framework* yang populer saat ini:

- a) *Eloquent ORM*: Memungkinkan interaksi dengan *Database* menjadi lebih intuitif dan efisien. Dengan *Eloquent*, pengembang dapat melakukan konfigurasi *Database* menggunakan sintaks yang mirip dengan bahasa pemrograman, sehingga mengurangi kerumitan penulisan *query SQL*.
- b) *Blade Templating Engine*: Digunakan untuk pembuatan antarmuka. *Blade* menawarkan sintaks yang sederhana dan rapi, memungkinkan pengembang untuk membuat template *HTML* yang dinamis dan mudah diatur. *Blade* juga mendukung inheritance template, sehingga dapat menciptakan struktur *layout* yang konsisten dan menghindari duplikasi kode.
- c) Migrasi: Memungkinkan pengembang untuk mendefinisikan perubahan skema *Database* menggunakan kode *PHP*, sehingga mempermudah pengelolaan versi dan kolaborasi dalam tim pengembang. Dengan migrasi, pengembang dapat melacak dan menerapkan perubahan skema *Database* dengan cara yang terstruktur dan dapat diulang.
- d) Dokumentasi Lengkap: *Laravel* menawarkan dokumentasi yang lengkap dan komunitas yang aktif, sehingga pengembang dapat dengan mudah menemukan solusi untuk masalah yang dihadapi selama proses pengembangan.

Database Management System (DBMS) atau sistem manajemen basis data adalah perangkat lunak yang digunakan untuk mengelola dan mengatur data dalam suatu basis data. Basis data merupakan kumpulan data yang terstruktur, tersimpan, dan dapat diakses dengan mudah. *MySQL* digunakan dalam proses pengembangan perangkat lunak ini.

MySQL merupakan sistem manajemen *Database* relasional yang telah lama digunakan dan memiliki kestabilan serta kinerja yang baik (Rizqi, 2023). Menurut Siregar et al., (2024), ada beberapa keunggulan *MySQL*, antara lain:

- a) Menyediakan sistem basis data berkecepatan tinggi yang sempurna untuk proyek-proyek kecil hingga menengah.
- b) *MySQL* dapat digunakan secara gratis dan bersifat open-source.
- c) Skalabilitas, konektivitas, dan keamanan yang baik menjadikan *MySQL* sistem manajemen *Database* yang populer digunakan saat ini, aman sekaligus fleksibel.

Untuk pengembangan di bagian aplikasi *mobile*, android studio digunakan sebagai pilihan. Android Studio merupakan sebuah *Integrated Development Environment (IDE)* khusus untuk membangun aplikasi yang berjalan pada platform android. Bahasa pemrograman utama yang digunakan adalah *Java*, sedangkan untuk membuat tampilan atau layout, digunakan bahasa *XML*. *Java* adalah bahasa pemrograman yang didesain agar dapat mengetik sintaks yang universal dan dapat digunakan di berbagai perangkat tanpa memerlukan kompilasi ulang. *Java* dikenal karena kesederhanaan dan kemudahannya yang menjadikannya pilihan populer untuk pengembangan aplikasi yang memiliki cakupan luas termasuk aplikasi *mobile*, aplikasi web, dan lain-lain (Martinez et al., 2023). *XML* adalah singkatan dari eXtensible Markup Language digunakan untuk membuat tampilan atau layout. *XML* didesain untuk mampu menyimpan data secara ringkas dan mudah diatur (Sibuea, 2022).

API pada perangkat lunak diperlukan untuk menjalankan fungsi dan fitur-fitur yang ditawarkan. *Application Programming Interface (API)* merupakan teknologi antarmuka yang dibangun oleh pengembang sistem supaya sebagian atau keseluruhan fungsi sistem dapat diakses secara bersamaan dengan baik (Triawan & Siboro, 2021). Dalam perangkat lunak yang penulis kembangkan, berikut adalah *API* yang digunakan untuk menjalankan fitur-fitur yang ditawarkan:

Daily Task mengelola tugas harian dengan fungsi utama:

- a. *index*: Menampilkan daftar tugas, dibedakan berdasarkan peran (karyawan hanya melihat tugasnya, manager melihat semua tugas buatannya).
- b. *store*: Menambahkan tugas baru, hanya dapat dilakukan oleh manager dengan validasi data (nama tugas, deskripsi, status, username karyawan, dan due date).
- c. *show*: Menampilkan detail tugas berdasarkan id.
- d. *update*: Memperbarui tugas dengan validasi input (status, nama tugas, deskripsi, due date).
- e. *destroy*: Menghapus tugas berdasarkan id.

Karyawan mengelola data karyawan dengan fungsi:

- index*: Menampilkan data karyawan non-manager dalam format JSON.
- import*: Mengimpor data karyawan dari file Excel dengan validasi.
- export*: Mengekspor data karyawan ke file Excel.
- destroy*: Menghapus data karyawan berdasarkan id dengan validasi keberadaan data.

Kolam mengelola data kolam dengan fungsi:

- index*: Menampilkan semua data kolam.
- store*: Menambah kolam baru dengan validasi input.
- show*: Menampilkan detail kolam berdasarkan id.
- update*: Memperbarui data kolam berdasarkan id.
- destroy*: Menghapus kolam berdasarkan id.

Modul mengelola data modul dengan fungsi:

- index*: Menampilkan semua data modul.
- store*: Menambah modul baru dengan validasi, termasuk gambar dan file opsional.
- show*: Menampilkan detail modul berdasarkan id.
- update*: Memperbarui data modul dan mengganti gambar/file jika ada.
- destroy*: Menghapus modul dan file terkait.

Pembelian Pakan mengelola data pembelian pakan dengan fungsi:

- index*: Menampilkan semua data pembelian pakan.
- store*: Menambah pembelian pakan baru dengan validasi dan memperbarui jumlah pakan di kolam.
- show*: Menampilkan detail pembelian pakan berdasarkan id.
- update*: Memperbarui data pembelian pakan dan memperbarui jumlah pakan di kolam.
- destroy*: Menghapus pembelian pakan dan mengembalikan jumlah pakan di kolam.

Penjualan Ikan mengelola data penjualan ikan dengan fungsi:

- index*: Menampilkan semua data penjualan ikan.
- store*: Menambah penjualan ikan baru dengan validasi dan memperbarui jumlah ikan di kolam.
- show*: Menampilkan detail penjualan ikan berdasarkan id.
- update*: Memperbarui data penjualan ikan dan memperbarui jumlah ikan di kolam.
- destroy*: Menghapus penjualan ikan dan mengembalikan jumlah ikan di kolam.

Berdasarkan metode pengembangan yang kita gunakan, adapun parameter yang penulis gunakan sebagai acuan kesuksesan pengembangan perangkat lunak.

Tabel 1. Perbedaan *role manager* dengan karyawan

Karyawan	Manager
Memperbarui data profil (seperti username, nama, email, no telepon, alamat, dan foto)	Memperbarui data profil (seperti username, nama, email).
Mengakses data yang berkaitan dengan tugas mereka.	Memiliki hak untuk mendaftarkan pengguna baru (termasuk karyawan).
Mengakses data yang berkaitan dengan tugas mereka.	Memiliki akses lebih luas untuk mengelola data pengguna dan melihat laporan.

Perangkat lunak yang penulis kembangkan membagi kegunaan berdasarkan *role*. Terdapat dua *role* yang diterapkan yaitu *manager* dan karyawan

Tabel 2. Parameter Penilaian Pengembangan Perangkat Lunak

No	Acuan Pengembangan	Nilai Kontribusi
1	Pengembangan mengikuti dan sesuai dengan arahan <i>timeline</i>	30%
2	Implementasi SDLC sesuai rancangan	30%
3	Implementasi <i>framework</i> dan <i>database</i>	40%

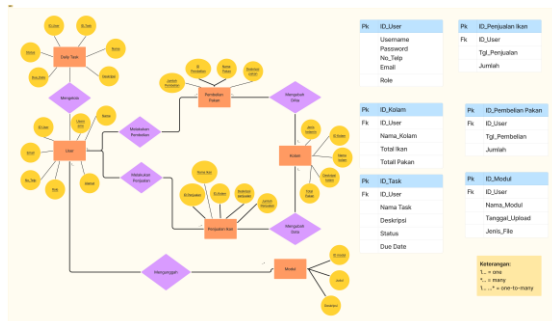
IV. HASIL DAN PEMBAHASAN

4.1 Planning

Dari hasil pengembangan yang kami lakukan, maka kami dapat menganalisa permasalahan yang ada pada manajemen tambak ikan. Pencatatan data masih dilakukan secara manual, baik dalam pencatatan data penjualan maupun data pemasukan. Proses pencatatan secara manual memerlukan tempat penyimpanan yang sangat luas akibat tumpukan kertas, serta membutuhkan waktu yang lama dalam proses pencarian kembali data. Akibatnya, pelayanan terhadap pelanggan menjadi tidak efisien dan berpotensi terjadi kesalahan karena waktu pencarian data yang lama. Oleh karena itu, kami mengusulkan ide dengan membuat pencatatan secara daring dengan fitur transaksi data masuk/keluar, produk, *daily task*, dan modul dengan nama *Fishseerlog*. Inisiasi ini diharapkan dapat memberikan efisiensi dalam proses pencatatan

tambak ikan. Di bawah ini merupakan diagram yang diusulkan untuk mendukung solusi tersebut.

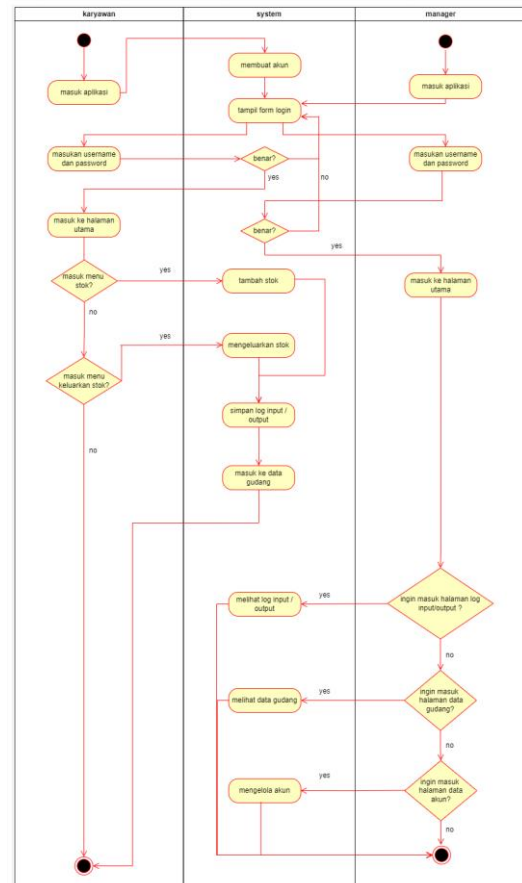
4.2. Design



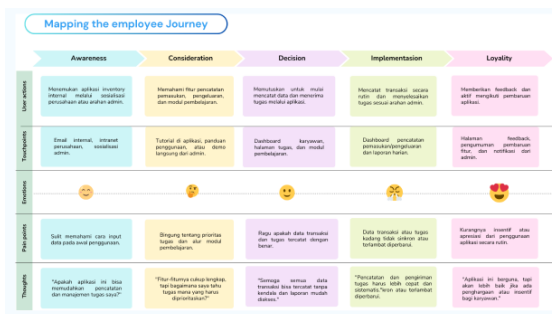
Gambar 3. ERD Fisheerlog

Berdasarkan gambar *Entity Relationship Diagram (ERD)*, sistem manajemen tambak ikan melibatkan beberapa entitas utama yang saling terhubung. *User* adalah entitas pusat yang menyimpan informasi pengguna seperti *id_User*, *Username*, dan *Role*. Setiap pengguna dapat memiliki beberapa *Daily task* yang harus diselesaikan, serta dapat melakukan Pembelian Pakan dan Penjualan Ikan.

Selain itu, *User* juga mengelola Kolam, yang berisi informasi tentang kolam ikan, serta *Modul* untuk edukasi karyawan. Semua entitas ini saling terhubung, memungkinkan sistem untuk mengelola data operasional tambak ikan secara efisien.



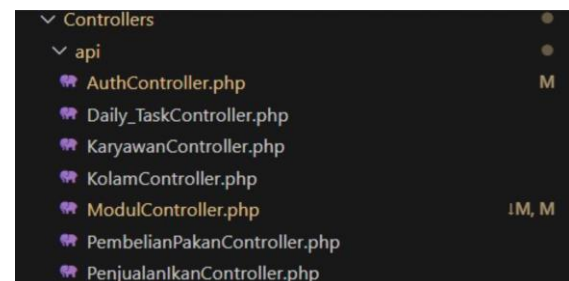
Gambar 5. User Journey



Gambar 4. User Journey

Pada bagian hasil dan pembahasan, kami menjelaskan tahapan pengembangan sistem yang dilakukan secara bertahap untuk memastikan setiap komponen berfungsi sesuai kebutuhan.

Tahap pertama adalah pengembangan *API* sebagai fondasi sistem, yang mencakup pembuatan endpoint untuk transaksi data masuk/keluar, manajemen produk, tugas harian (*daily task*), dan modul pembelajaran. *API* dirancang untuk mendukung komunikasi antara frontend dan database, serta diuji menggunakan alat seperti Postman untuk memastikan keandalannya.



Gambar 6. API

Setelah *API* selesai, tahap berikutnya adalah pengembangan *frontend* untuk antarmuka pengguna. Proses ini melibatkan pembuatan desain yang

responsif dan intuitif, sehingga pengguna dapat dengan mudah mengakses fitur seperti pencatatan data, dan pengelolaan produk. *Frontend* dirancang menggunakan framework modern yang mendukung pengintegrasian dengan *API*, memungkinkan data ditampilkan secara *real-time* dan interaktif.

Melalui pendekatan bertahap ini, sistem yang dikembangkan diharapkan mampu memberikan solusi yang efisien dan efektif untuk manajemen tambak ikan.

4.3. Alur Sistem

Ilustrasi berikut menggambarkan proses kerja aplikasi **Fisheerlog** dalam membantu pengguna mengelola budidaya ikan. Aplikasi ini dirancang untuk mendukung pencatatan transaksi, pengelolaan stok, edukasi karyawan, dan distribusi tugas harian dalam budidaya ikan.



Gambar 7. Ilustrasi Alur Sistem

4.4. Development

Tahap ini melibatkan pengujian aplikasi *Fisheerlog* sebelum dirilis kepada pengguna. Tujuan utama dari tahap ini adalah memastikan bahwa semua fitur aplikasi berjalan sesuai dengan rancangan dan bebas dari bug atau gangguan yang dapat menghambat pengalaman pengguna.

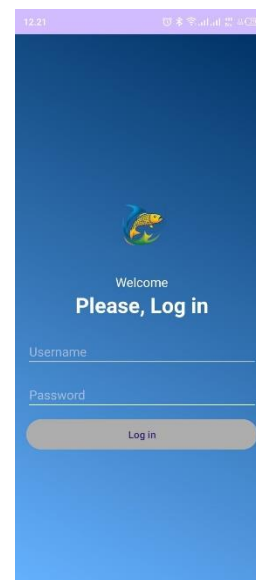
1. Role Pengguna

1. Login ke Aplikasi

Pengguna memulai dengan masuk ke aplikasi *Fisheerlog* menggunakan akun terdaftar. Setelah berhasil masuk, pengguna diarahkan ke halaman utama yang menampilkan berbagai menu fitur.



Gambar 8. Splash Screen



Gambar 9. Tampilan Login

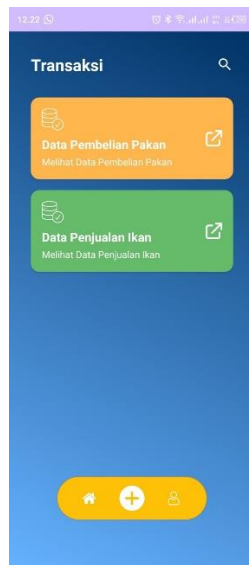
2. Pencatatan Transaksi

a. Transaksi Keluar Ikan:

Pengguna memilih menu transaksi keluar ikan untuk mencatat data penjualan ikan. Data yang dimasukkan mencakup jumlah ikan yang dijual, harga, dan waktu transaksi.

b. Transaksi Data Masuk:

Untuk pembelian pakan ikan, pengguna memilih menu transaksi data masuk dan mencatat detail pembelian, seperti jenis pakan, jumlah, dan tanggal.



Gambar 10. Tampilan Data masuk/keluar

3. Pengelolaan Stok Kolam

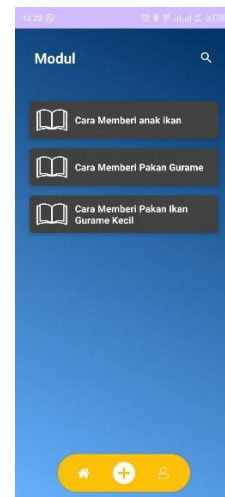
Pengguna dapat memantau stok kolam melalui fitur ini. Data real-time tentang jumlah ikan di setiap kolam dan ketersediaan pakan ditampilkan. Fitur ini membantu pengguna menentukan kebutuhan operasional, seperti waktu pemberian pakan atau jadwal panen.



Gambar 11. Tampilan Data Kolam

4. Edukasi Karyawan

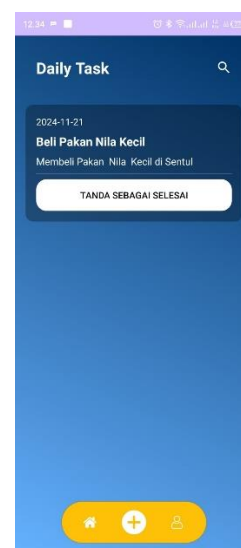
Pengguna dapat mengakses modul edukasi yang berisi materi tentang pengelolaan kolam, teknik budidaya ikan, dan informasi terbaru tentang teknologi budidaya. Karyawan dapat mempelajari modul ini untuk meningkatkan keterampilan mereka.



Gambar 12. Tampilan Modul

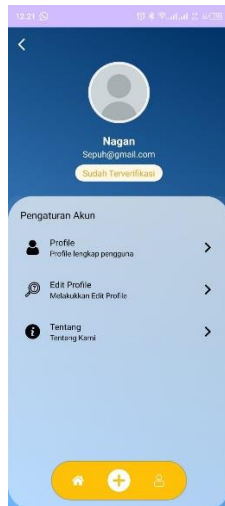
5. Distribusi Tugas Harian

Melalui fitur *daily task*, pengguna dapat mengatur tugas-tugas harian karyawan, seperti pemberian pakan, pembersihan kolam, atau pemeriksaan kondisi ikan. Tugas yang telah selesai dapat dilaporkan melalui aplikasi.

Gambar 13. Tampilan *Daily task*

6. Pengaturan Keamanan

Pengguna dapat mengubah *profile* seperti Gambar, *Username*, *Email*, Nomor telepon, dan Alamat mereka.



Gambar 14. Tampilan Pengaturan

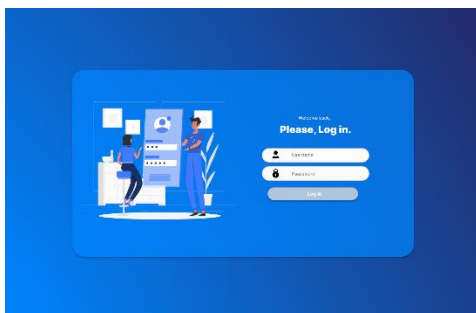


Gambar 15. Tampilan Pengaturan

2. Role Admin

1. Login Dashboard

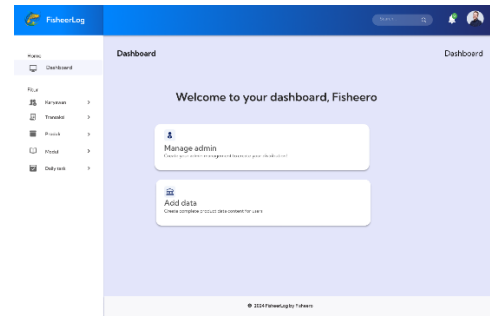
Tampilan login yang ditampilkan yaitu ada username dan juga kata sandi dimana pengguna dapat mengisi username dan kata sandi yang digunakan.



Gambar 16. Tampilan Login Admin

2. Dashboard

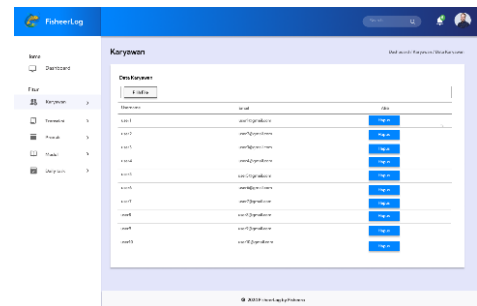
Dashboard halaman utama ketika login dan bisa diarahkan ke fitur lainnya.



Gambar 17. Dashboard

3. Pendataan Karyawan

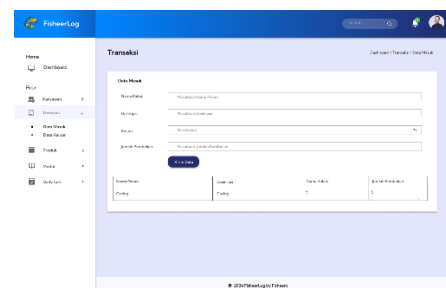
Admin (Manager) mendata karyawan yang bekerja di perusahaannya secara daring.



Gambar 18. Tampilan Data Karyawan

4. Transaksi Masuk/Keluar

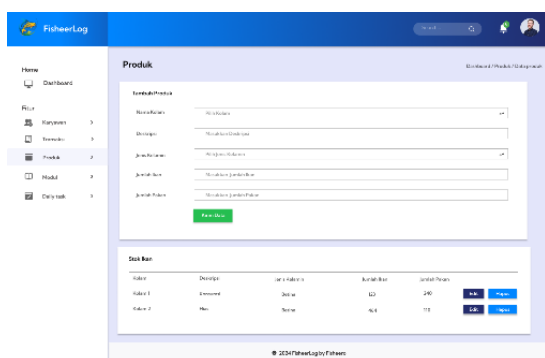
Admin bisa mengubah dan menambahkan data transaksi yang sudah dimasukkan oleh karyawan melalui Aplikasi.



Gambar 19. Tampilan Transaksi

5. Data Produk

Admin dapat menambahkan atau mengedit data produk baru melalui fitur transaksi yang tersedia. Selain itu, admin juga dapat mengelola data stok ikan yang telah dimasukkan sebelumnya oleh karyawan, termasuk memperbarui atau menghapus data melalui tabel yang tersedia.



Gambar 20. Tampilan Produk

6. Data Modul

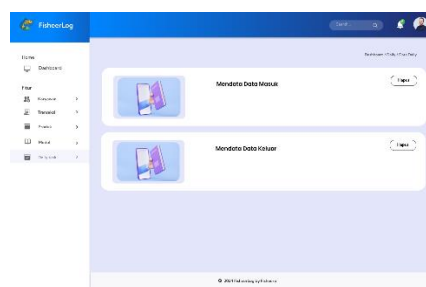
Admin dapat melihat daftar modul yang telah ditambahkan, seperti materi penjelasan mengenai aplikasi *Fisheerlog* atau panduan tentang pakan dan jenis yang diperlukan. Setiap modul ditampilkan dengan ilustrasi dan judul yang menggambarkan isi modul. Admin juga memiliki kemampuan untuk menghapus modul yang tidak lagi relevan melalui tombol Hapus yang tersedia di setiap modul.



Gambar 21. Tampilan Modul

7. Daily task

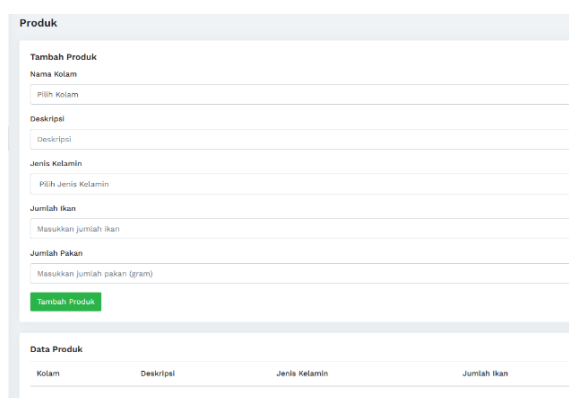
Halaman ini menampilkan daftar tugas harian yang perlu dikelola oleh admin, seperti mendata data masuk dan mendata data keluar. Setiap tugas dilengkapi dengan ilustrasi dan deskripsi singkat. Admin dapat menghapus tugas yang sudah tidak relevan melalui tombol Hapus yang tersedia pada setiap tugas. Halaman ini bertujuan untuk mempermudah pengelolaan aktivitas harian dalam aplikasi *Fisheerlog*.



Gambar 22. Tampilan Modul

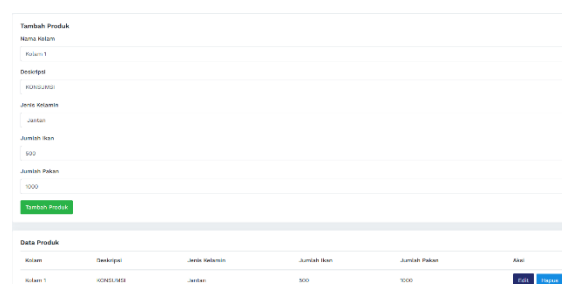
4.5 Testing

Pada tahap ini perangkat lunak di uji coba keberhasilan fungsinya. Pengujian dengan melakukan aksi sebagai *manager* di *web* dengan menggunakan fitur tambah kolom.



Gambar 23. Tampilan Data Kosong

Pada uji coba ini penulis memasuki lama web sebagai *manager*, lalu memilih fitur kolam pada sidebar yang tersedia. Tampilan ini menunjukkan data kolam yang tersedia yang saat ini masih kosong. Selanjutnya tombol tambah kolam di klik lalu akan muncul kolom yang harus diisi.



Gambar 24. Tampilan Data Kosong

Terdapat kolom a,b,c,d yang harus diisi sebagai keperluan *data*. Setelah semua kolom diisi, maka tombol tambah data di klik. Setelah itu data akan muncul sebagai baris baru di tabel kolam



Gambar 25. Tampilan Data Kosong

4.6 Deployment

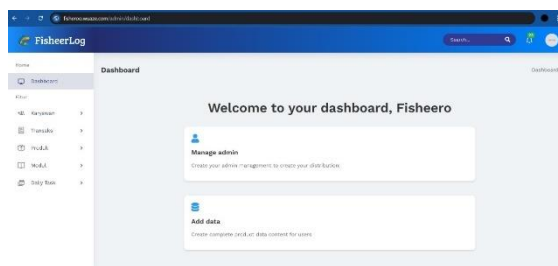
Pada tahap ini perangkat lunak sudah di uji coba kelayakan fungsi dan fitur-fiturnya dan mendapat respon positif berfungsi dari pengujian tersebut. Tahap selanjutnya adalah melakukan penyebaran dengan memberikan *domain* sekaligus *hosting* perangkat lunak. Untuk layanan *domain* dan *hosting* dalam pengembangan perangkat lunak *Fisheerlog* adalah *infinityfree* yang memberikan layanan *hosting* dan *domain* secara gratis.



File	Size	Upload Date	Status
assets		Dec 1, 2024	uploaded
build		Dec 1, 2024	uploaded
images		Dec 1, 2024	uploaded
laravel		Dec 2, 2024	uploaded
modul		Dec 1, 2024	uploaded
intaccess	603B	Dec 1, 2024	uploaded
fawicomico	GB	Dec 1, 2024	uploaded
index.php	480B	Dec 1, 2024	uploaded
robots.txt	24B	Dec 1, 2024	uploaded

Gambar 26. Upload file untuk hosting

Setelah proses *upload* selesai, maka *web* sudah bisa diakses melalui laman <http://fisherloo.wuaze.com>.



Gambar 27. Tampilan *web* setelah di *hosting* dan diakses melalui *domain*

4.7. Review

Setelah perangkat lunak melewati semua tahap yang diperlukan untuk penyempurnaan, perangkat lunak diperiksa secara berkala dan melihat hasil dari pengembangannya. Parameter kesuksesan perangkat lunak ditinjau ulang untuk melihat apakah hasil sudah sesuai dengan tujuan dan permasalahan yang dihadapi. Setelah melewati proses peninjauan ulang, perangkat lunak *Fisheerlog* memiliki beberapa kekurangan seperti tidak adanya data waktu untuk transaksi sehingga *data* tidak dihasilkan akurat dengan waktu. Penambahan akun masih secara *manual* melalui pemanggilan *API* dan belum dikonfigurasi fungsinya dalam perangkat lunak.

Hasil dari peninjauan ini akan di kaji lebih lanjut agar menghasilkan produk yang dapat memuaskan klien dan dapat mengatasi masalah dengan maksimal.

V. KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil dan pembahasan yang dipaparkan, terdapat beberapa kesimpulan mengenai pengembangan perangkat lunak *Fisheerlog*:

1. Menetapkan timeline pra-pengembangan mempermudah dalam penggambaran dan penyesuaian proses pengembangan, namun terdapat beberapa ketidaksesuaian waktu pengerjaan namun tidak berdampak besar pada kelancaran pengembangan.
2. Pengembangan perangkat lunak menggunakan metode *SDLC* dan model *Agile* mempermudah proses pengembangan dengan memaksimalkan setiap tahap dan meminimalisir resiko.
3. Implementasi dari pengembangan menggunakan *framework laravel* dengan bahasa pemrograman *PHP* dan *MySQL* sebagai *database* mempermudah pengembangan perangkat lunak yang dinamis dan mempersingkat waktu pengembangan dengan fitur migrasi yang mempercepat konfigurasi skema *databas*.

Saran

Berdasarkan kesimpulan yang diuraikan, terdapat beberapa saran yang dapat diajukan untuk pengembangan lebih lanjut *Fisheerlog*:

1. Meningkatkan fungsionalitas fitur modul agar dapat mengakses materi melalui video agar lebih mudah dipahami.
2. Menambahkan catatan tanggal untuk fitur transaksi untuk menghasilkan data yang akurat.
3. Menambahkan siste *IoT* pada kolam untuk memonitor kondisi ikan sekaligus kolam agar lebih terjaga kualitasnya.

DAFTAR PUSTAKA

- Adani, M. R. (2023). Mengenal metode Agile dan 12 prinsipnya bagi software.
- Amartya Dewa, S. B., & Azizah, N. L. (2024). Perancangan sistem informasi sekolah berbasis web menggunakan metode *SDLC*. Retrieved from <https://archive.umsida.ac.id/index.php/archive/p reprint/download/4065/28825/32515>
- Deputi Bidang Koordinasi Sumber Daya Maritim. (2020-2024). Dokumen rencana strategis (Renstra) Deputi Bidang Koordinasi Sumber Daya Maritim. Retrieved from <https://maritim.go.id/uploads/magazine/20230712114619-2023-07-12magazine114617.pdf>
- Dewi, T. T. (2023). Manfaat dan tantangan pengembangan e-modul pada pendidikan kejuruan: Sebuah kajian literatur.
- Direktorat Pengelolaan Sumber Daya Ikan. (2024). Laporan kinerja Direktorat Pengelolaan Sumber

- Daya Ikan pada Triwulan II Tahun 2024. Retrieved from <https://kkp.go.id/download-pdf-akuntabilitas-kinerja/akuntabilitas-kinerja-pelaporan-kinerja-laporan-kinerja-direktorat-pengelolaan-sumber-daya-ikan-triwulan-ii-tahun-2024.pdf>
- Kalsum, L., et al. (2021). Pengolahan air payau menjadi air bersih menggunakan metode elektrokoagulasi.
- Kementerian Kelautan dan Perikanan. (2023). Laporan kinerja KKP tahun 2023. Retrieved from https://kkp.go.id/storage/public/AkuntabilitasKinerja/document-akuntabilitas-kinerja-664ec554a0cccBuku_LAPORAN_KKP_2023_final_HR%20ok.pdf
- Kusumah, R. M., & Penangsang, P. (2024). Analisis kelayakan usaha tambak ikan nila di Desa Prasung, Kecamatan Buduran, Kabupaten Sidoarjo.
- Nurfadilah, N. (2022). Pemberdayaan petani tambak ikan bandeng pada masa pandemi di Kecamatan Pangkajene oleh Dinas Perikanan Kabupaten Pangkajene dan Kepulauan Provinsi Sulawesi Selatan. Retrieved from <http://eprints.ipdn.ac.id/8416/1/NidA%20NURFADHILAH%20KELAS%20C1%20PEPM.pdf>
- Paksi, A. B., Hafidhoh, N., & Bimonugroho, S. K. (2023). Perbandingan model pengembangan perangkat lunak untuk proyek tugas akhir program vokasi.
- Rizqi, A. R. (2023). Studi komparasi database MySQL, MongoDB, dan InfluxDB sebagai data storage sensor berbasis IoT.
- Sibuea, S., Saputro, M. I., Annan, A., & Widodo, Y. B. (2022). Aplikasi mobile collection berbasis Android pada PT. Suzuki Finance Indonesia. Retrieved from <https://journal.amikveteran.ac.id/index.php/jitek/article/download/185/136/817>
- Sinlae, F., Irwanda, E., Maulana, Z., & Syahputra, V. E. (2024). Penggunaan framework Laravel dalam membangun aplikasi website berbasis PHP.
- Siregar, U. K., Sitakar, T. A., Haramain, S., Lubis, Z. N. S., Nadhirah, U., & Yahfizham. (2024). Pengembangan database management system menggunakan MySQL.
- Sukma, D. (2019). Optimalisasi usaha budidaya tambak ikan bandeng (Studi kasus: Tambak Desa Keude Bireum, Kecamatan Bireum Bayeun, Kota Langsa).
- Triawan, Anggra & Siboro, Andika. (2021). Penerapan Application Programming Interface (API) Pada Push Notification Untuk Informasi Monitoring Stok Barang Minim. Teknois : Jurnal Ilmiah Teknologi Informasi dan Sains. 11. 107-114. 10.36350/jbs.v11i2.120.