



BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Aplikasi

Aplikasi merupakan program komputer yang tersusun dari sejumlah instruksi dan perintah yang dirancang untuk memungkinkan komputer memproses data masukan menjadi keluaran. Selain itu, aplikasi juga dapat didefinisikan sebagai kumpulan berkas terintegrasi yang bekerja sama untuk menjalankan fungsi atau tugas spesifik dijelaskan oleh Wulandari et al., (2021).

Aplikasi merupakan bagian dari perangkat lunak yang dibuat untuk menyelesaikan permasalahan tertentu yang dihadapi pengguna dengan memanfaatkan fungsi dan kemampuan komputer dijelaskan oleh Hendriyati & Yusta, (2021).

2.1.2 Penentuan Komoditas

Penentuan komoditas unggulan merupakan langkah penting dalam mendukung pembangunan sektor pertanian dan peningkatan pertumbuhan ekonomi daerah. Komoditas unggulan di suatu wilayah dapat ditentukan berdasarkan tingkat produksi serta kemampuan komoditas tersebut dalam memenuhi kebutuhan wilayah sendiri maupun wilayah lain dijelaskan oleh Wika Asмага et al., (2025)

Penetapan komoditas potensial merupakan salah satu strategi pemerintah daerah dalam meningkatkan produktivitas sektor ekonomi serta mendorong pertumbuhan pembangunan wilayah secara berkelanjutan. Komoditas potensial diartikan sebagai komoditas yang memiliki peluang untuk dikembangkan menjadi komoditas unggulan daerah di masa depan dijelaskan oleh Suri et al., (2024)

2.1.3 Perikanan Tangkap

Kegiatan perikanan tangkap pada dasarnya merupakan usaha pemanfaatan sumber daya ikan melalui aktivitas penangkapan di perairan menggunakan alat tangkap tertentu dengan tujuan memperoleh hasil produksi yang bernilai ekonomi. Pendapatan usaha perikanan tangkap diperoleh dari selisih antara nilai produksi hasil tangkapan dengan

total biaya operasional yang dikeluarkan selama kegiatan penangkapan berlangsung (Rahim, 2011 dalam Arumtyas et al., 2023)

Selain itu, perikanan tangkap juga memiliki peranan penting dalam mendukung pembangunan ekonomi nasional, namun dalam pelaksanaannya menghadapi berbagai tantangan seperti keterbatasan sumber daya ikan, dominasi nelayan skala kecil, serta ancaman praktik penangkapan ikan yang tidak terlaporkan yang dapat mempengaruhi keberlanjutan produksi perikanan. Oleh karena itu, pengelolaan perikanan tangkap perlu dilakukan secara terencana dan berbasis data agar pemanfaatan sumber daya ikan tetap lestari dan memberikan manfaat optimal bagi masyarakat (Kusdiantoro et al., 2019).

2.1.4 Perairan Darat

Perairan darat merupakan wilayah perairan yang berada di kawasan daratan seperti sungai, danau, rawa, waduk, dan badan air lainnya yang menjadi habitat bagi berbagai jenis organisme akuatik. Perairan darat memiliki potensi sumber daya ikan yang cukup besar serta berperan penting dalam mendukung ketahanan pangan, penyediaan sumber protein, dan peningkatan ekonomi masyarakat yang bergantung pada sektor perikanan (Nurseptiani et al., 2021).

Selain itu, perairan darat juga memiliki fungsi ekologis dan ekonomi yang penting bagi masyarakat, terutama dalam mendukung kegiatan perikanan tangkap skala kecil serta menjaga keseimbangan ekosistem perairan. Namun dalam pengelolaannya, perairan darat menghadapi berbagai tantangan seperti penurunan kualitas lingkungan, pencemaran perairan, serta eksploitasi sumber daya ikan yang tidak terkendali. Oleh karena itu, pengelolaan perairan darat perlu dilakukan secara berkelanjutan dan berbasis data agar potensi sumber daya ikan yang tersedia dapat dimanfaatkan secara optimal serta tetap menjaga kelestarian ekosistem perairan (Gulo, 2025).

2.1.5 Metode Rule-Based System

Setiap aturan (*rule*) dalam *Rule-Based System* tersusun atas dua bagian utama, yaitu bagian *IF* yang berfungsi sebagai kumpulan fakta atau kondisi, serta bagian *THEN* yang berperan sebagai kesimpulan atau tindakan. Secara umum, sebuah *rule* dapat memiliki

lebih dari satu fakta yang dihubungkan menggunakan operator logika *AND*, *OR*, atau kombinasi dari keduanya (Ardiansyah, 2023). Secara konseptual, *Rule Based System* terdiri atas tiga komponen utama, yaitu *Knowledge Base*, *Inference Engine*, dan *Working Memory*. *Knowledge Base* berisi kumpulan aturan *IF-THEN* yang menjadi dasar pengetahuan sistem. *Inference Engine* berfungsi untuk mencocokkan data masukan dengan aturan yang tersedia melalui mekanisme penalaran logis secara berulang hingga diperoleh kesimpulan. Sementara itu, *Working Memory* digunakan untuk menyimpan fakta atau data sementara yang berasal dari pengguna selama proses pengambilan keputusan berlangsung (Sunarto et al., 2025).

Pemenuhan aturan dalam Rule-Based System dapat direpresentasikan pada persamaan (1) sebagai berikut:

$$\sum_{i=1}^n s_i \quad (1)$$

Keterangan:

- $s_i = 1$ jika aturan ke- i terpenuhi
- $s_i = 0$ jika aturan ke- i tidak terpenuhi
- n adalah jumlah aturan

2.1.6 Unified Modelling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan perangkat lunak yang telah distandardisasi dan digunakan sebagai sarana untuk menyusun cetak biru (*blueprint*) pengembangan perangkat lunak. UML dapat dimanfaatkan untuk melakukan visualisasi, spesifikasi, konstruksi, serta dokumentasi berbagai bagian sistem dalam pengembangan perangkat lunak (Sumiati et al., 2021). Dalam penerapannya, metode ini mendukung proses analisis dan desain perangkat lunak secara iteratif, yang mencakup identifikasi kelas dan objek, pemahaman hubungan semantik antar objek dan kelas, serta perincian antarmuka dan implementasinya (Voutama et al., 2022).

2.1.7 Use Case Diagram

Use Case Diagram merupakan diagram yang digunakan untuk menggambarkan interaksi antara sistem dengan pengguna atau sistem eksternal. Diagram ini secara grafis menunjukkan aktor yang terlibat serta bentuk interaksi yang diharapkan antara aktor dan sistem. *Use Case Diagram* digunakan sebagai model perilaku (*behavior*) sistem Informasi yang akan dikembangkan dengan mendeskripsikan hubungan interaksi antara satu atau lebih aktor dengan sistem (Oktavianto, 2022). Dalam hal ini, *Use case* digunakan untuk menggambarkan kebutuhan sistem dari sudut pandang pengguna secara sederhana dan mudah dipahami. *Use case* juga merepresentasikan fungsi-fungsi yang disediakan oleh sistem, di mana setiap fungsi berperan sebagai unit interaksi yang memungkinkan terjadinya pertukaran pesan antara aktor dengan sistem maupun dengan unit lainnya (Amanda & Nopriani, 2024). Berikut simbol *Use Case Diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol-simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Use Case</i>	Menunjukkan fungsi sistem yang digambarkan sebagai unit fungsional yang saling bertukar pesan dengan aktor.
2.		<i>Actor</i>	Merupakan pihak luar (manusia, sistem, atau proses) yang berinteraksi dengan sistem.
3.		<i>Association</i>	Hubungan komunikasi antara aktor dan <i>use case</i> . Menunjukkan partisipasi atau interaksi langsung dalam skenario sistem.

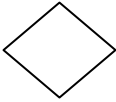
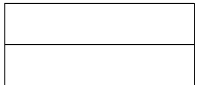
4.		<i>Extend</i>	Hubungan antara <i>use case</i> tambahan dan induk. Bersifat opsional dan dapat berjalan sendiri tanpa induk.
5.		<i>Uses</i>	Relasi antara <i>use case</i> utama dan <i>use case</i> yang selalu dibutuhkan. <i>Use case</i> tambahan akan selalu dipanggil oleh <i>use case</i> utama atau menjadi syarat sebelum eksekusi.
6.		<i>Generalization</i>	Hubungan antara <i>use case</i> umum dan khusus. Menunjukkan fungsi dan lainnya

Sumber: (Yuliawati, 2024)

2.1.8 Activity Diagram

Activity Diagram digunakan untuk menggambarkan alur kerja (*workflow*) atau rangkaian aktivitas yang terjadi dalam suatu sistem, proses bisnis, maupun menu yang terdapat pada perangkat lunak. Diagram ini berfokus pada aktivitas yang dilakukan oleh sistem, bukan pada tindakan aktor, sehingga menggambarkan proses yang dapat dijalankan oleh sistem secara keseluruhan (Aditya & Pranatawijaya, 2021). Selain itu, *Activity Diagram* menunjukkan berbagai alur aktivitas dalam sistem yang sedang dirancang, mulai dari titik awal proses, kemungkinan alur yang dapat terjadi, hingga akhir proses. Diagram ini juga menggambarkan proses-proses yang dapat berlangsung pada beberapa skenario eksekusi, sehingga memberikan gambaran umum mengenai jalur aktivitas pada tingkat yang lebih tinggi tanpa menampilkan perilaku internal sistem secara detail (Hasanah & Untari, 2021). Berikut simbol *Activity Diagram* dapat dilihat pada Tabel 2.2

Tabel 2.2 Simbol-simbol *Activity Diagram*

No.	Nama	Simbol	Keterangan
1.	<i>Start Point</i>		Menyatakan bahwa sebuah objek dimulai atau diawali dari titik ini.
2.	<i>Activity</i>		Menyatakan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
3.	<i>Decision</i>		Menunjukkan penggambaran suatu keputusan atau tindakan yang harus diambil berdasarkan kondisi tertentu.
4.	<i>Join</i>		Menyatakan untuk menggabungkan kembali <i>activity</i> atau <i>action</i> yang berjalan secara paralel.
5.	<i>Swimlane</i>		Mengelompokkan aktivitas aktor atau dalam sistem, dan untuk menunjukkan yang bertanggung jawab.
6.	<i>End Point</i>		Menyatakan bahwa sebuah objek dihentikan atau diakhiri pada titik ini.

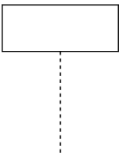
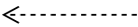
Sumber: (Yuliawati, 2024)

2.1.9 *Sequence Diagram*

Sequence Diagram merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antar objek dalam suatu sistem secara kronologis. Diagram ini menunjukkan bagaimana objek-objek saling berkomunikasi melalui pertukaran pesan yang terjadi secara berurutan selama eksekusi suatu skenario atau proses (Yuliawati, 2024). Selain itu, *Sequence Diagram* digunakan untuk menjelaskan perilaku sistem pada suatu skenario tertentu dengan menggambarkan hubungan interaksi antara entitas dan sistem, termasuk pesan yang dikirim dan diterima antar objek. Seluruh pesan ditampilkan

berdasarkan waktu eksekusi, sehingga diagram ini mampu menunjukkan waktu hidup objek serta alur komunikasi yang terjadi dalam sebuah *use case* secara jelas (Aditya & Pranatawijaya, 2021). Berikut simbol *Sequence Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol-simbol *Sequence Diagram*

No.	Nama	Simbol	Deskripsi
1.	<i>Entity Class</i>		Representasi sistem yang menjadi dasar perancangan struktur basis data.
2.	<i>Boundary Class</i>		Berfungsi mengelola batas komunikasi antara sistem dengan lingkungan eksternal atau aktor.
3.	<i>Object Lifeline</i>		Garis vertikal putus-putus yang menggambarkan keberadaan objek selama interaksi.
4.	<i>Message</i>		Mewakili komunikasi atau pertukaran interaksi
5.	<i>Message Return</i>		Menunjukkan arah umpan balik dalam rangkaian komunikasi atau alur interaksi objek.

Sumber: (Yuliawati, 2024)

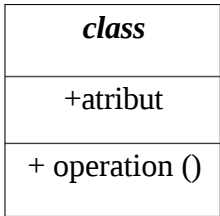
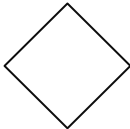
2.1.10 Class Diagram

Class Diagram merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan struktur sistem yang dibangun dari kumpulan kelas-kelas. Diagram ini menampilkan kelas-kelas yang terdapat dalam sistem beserta hubungan logis antar kelas tersebut, sehingga memberikan gambaran mengenai struktur statis sistem secara keseluruhan

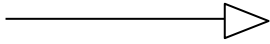
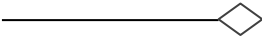
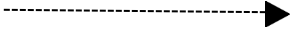
(Oktavianto, 2022). *Class Diagram* juga digunakan untuk menunjukkan paket, atribut, metode, serta relasi antar kelas yang ada dalam sistem.

Melalui diagram ini, dapat dipahami bagaimana kelas-kelas saling berinteraksi dan berhubungan secara logis. Dalam implementasinya, *Class Diagram* dapat mencakup kelas utama (*main class*), kelas antarmuka yang menangani tampilan dan validasi sistem, kelas yang merepresentasikan proses *use case* seperti *login*, serta kelas koneksi basis data yang berfungsi untuk mengelola akses dan proses *query* terhadap basis data (Aditya & Pranatawijaya, 2021). Berdasarkan uraian tersebut, *Class Diagram* digunakan untuk menggambarkan struktur statis sistem melalui pemodelan kelas, atribut, metode, dan relasi antar kelas, sehingga memudahkan pemahaman susunan dan hubungan komponen sistem secara keseluruhan. Berikut simbol *Use Case Diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol-simbol *Class Diagram*

No	Simbol	Deskripsi
1.	<i>Class</i> 	Kelas merupakan sekumpulan objek yang memiliki atribut dan operasi, di mana atribut merepresentasikan karakteristik perilaku objek dalam sistem.
2.	<i>Nary Association</i> 	Relasi di mana objek turunan (anak) mewarisi sifat dan struktur dari objek induknya.
3.	<i>Association</i> 	Relasi umum yang menghubungkan antara satu objek dengan objek lainnya.



4.	<i>Generalization</i> 	Relasi di mana objek turunan (anak) mewarisi sifat dan struktur dari objek induknya.
5.	<i>Aggregation</i> 	Hubungan antar kelas yang menunjukkan keterkaitan antara keseluruhan dan bagian-bagiannya (<i>whole-part relationship</i>).
6.	<i>Dependency</i> 	Hubungan yang menunjukkan perubahan pada suatu elemen independen dapat mempengaruhi elemen lainnya yang bergantung.

Sumber : (Yuliawati, 2024)

2.1.11 Kamus Data

Kamus data (*data dictionary*) merupakan sebuah katalog terstruktur yang berisi kumpulan daftar elemen data yang mengalir pada suatu sistem perangkat lunak. Pembuatan kamus data ini bertujuan untuk memberikan standardisasi penulisan agar setiap masukan (*input*) dan keluaran (*output*) informasi memiliki definisi yang seragam, jelas, dan dapat dipahami secara umum oleh seluruh pihak pengembang sistem, sebagaimana yang dikemukakan dalam pedoman perancangan Sukamto dan Shalahuddin (Dwiyana Liksha, 2021).

Secara arsitektural, dokumen ini berfungsi sebagai referensi pusat yang menjabarkan secara rinci metadata dari setiap entitas, tabel, dan atribut yang ada di dalam basis data. Keberadaan kamus data sangat esensial dalam tahapan perancangan guna meminimalisasi redudansi atau ambiguitas penamaan tipe data, batasan ukuran (*length*), maupun relasi kunci (*keys*) antar-tabel, sehingga integritas dan keamanan pengelompokan struktur penyimpanan program dapat selalu terjaga dengan baik (Lestika Atimi et al., 2025).



Penyusunan kamus data pada penelitian ini diimplementasikan sebagai dasar dalam merancang struktur basis data pada aplikasi penentuan komoditas perikanan tangkap perairan darat di Dinas Kelautan dan Perikanan Provinsi Sumatera Selatan. Standardisasi metadata ini digunakan untuk memetakan seluruh atribut data yang diperlukan dalam sistem, seperti data produksi perikanan dari berbagai kabupaten/kota, jenis komoditas ikan, serta kriteria penilaian yang digunakan dalam metode *Rule-Based System*. Dengan adanya kamus data, setiap elemen data yang digunakan dalam proses pengolahan dan analisis dapat terdokumentasi dengan jelas sehingga meminimalkan kesalahan dalam pengelolaan basis data sistem.

Tabel 2.1 Simbol-simbol pada Kamus Data

No	Simbol	Keterangan
1.	=	Disusun atau terdiri dari
2.	+	Dan
3.	[]	Baik...atau....
4.	{ } ⁿ	n kali diulang/bernilai banyak
5.	()	Data opsional
6.	*...*	Batas komentar

(Dwiwana Liksha, 2021)

2.1.12 PHP

Hypertext Preprocessor (PHP) merupakan bahasa pemrograman skrip yang beroperasi pada sisi peladen (*server-side scripting*) dan didesain secara spesifik untuk memfasilitasi pengembangan platform berbasis *website*. Bahasa pemrograman ini memiliki peran sentral dalam menghasilkan halaman web yang dinamis, di mana kode instruksi akan dieksekusi terlebih dahulu di dalam peladen sebelum hasil komputasinya ditransmisikan kepada peramban pengguna dalam format keluaran web standar (Raihani, 2025a).

Secara arsitektural, PHP memiliki keunggulan komparatif berupa kompatibilitas lintas platform serta kemudahan integrasi dengan berbagai



sistem manajemen basis data relasional, seperti MySQL. Sifat distribusinya yang sumber terbuka (*open-source*) memungkinkan pengembang perangkat lunak untuk membangun struktur informasi yang kompleks secara efisien, serta memfasilitasi pertukaran instruksi logika program skala besar secara komprehensif tanpa terikat oleh lisensi komersial (Fatih, 2025).

Pemilihan bahasa pemrograman PHP pada penelitian ini digunakan sebagai dasar pengembangan sistem penentuan komoditas perikanan tangkap perairan darat berbasis website pada Dinas Kelautan dan Perikanan Provinsi Sumatera Selatan. PHP berperan dalam mengelola proses pengolahan data produksi perikanan yang tersimpan dalam basis data serta menjalankan logika sistem dalam menerapkan metode *Rule-Based System*. Melalui kemampuan pemrosesan pada sisi peladen, PHP memungkinkan sistem untuk menganalisis data produksi perikanan dan menghasilkan informasi mengenai komoditas perikanan yang berpotensi secara dinamis melalui antarmuka website

2.1.13 Laravel

Laravel merupakan kerangka kerja (*framework*) berbasis bahasa pemrograman PHP yang bersifat sumber terbuka (*open-source*) dan dirancang khusus untuk memaksimalkan efisiensi pengembangan aplikasi web tingkat lanjut. Kerangka kerja ini mengadopsi arsitektur *Model-View-Controller* (MVC) yang secara sistematis memisahkan logika pemrosesan data, antarmuka visual, dan rute pengendalian sistem, sehingga proses penulisan kode menjadi lebih terstruktur dan mudah dikelola dalam skala perangkat lunak yang kompleks (Rahmawati & Sumarsono, 2024).

Secara fungsional, penggunaan kerangka kerja ini memfasilitasi pengembang melalui penyediaan berbagai pustaka bawaan (*built-in libraries*) yang komprehensif. Fitur-fitur esensial seperti sistem autentikasi keamanan, pemetaan relasi objek basis data atau *Object-Relational Mapping* (ORM) yang presisi, serta mesin templat (*template engine*) untuk penyajian antarmuka dinamis, secara signifikan



mempercepat siklus pengembangan perangkat lunak tanpa mengorbankan stabilitas maupun keamanan aplikasi (Valensyah & Irnawati, 2024)

Implementasi kerangka kerja Laravel pada penelitian ini digunakan sebagai fondasi utama dalam pembangunan aplikasi penentuan komoditas perikanan tangkap perairan darat berbasis website pada Dinas Kelautan dan Perikanan Provinsi Sumatera Selatan. Penerapan arsitektur Model-View-Controller (MVC) pada Laravel dimanfaatkan untuk memisahkan pengelolaan data produksi perikanan, logika analisis menggunakan metode *Rule-Based System*, serta penyajian antarmuka pengguna. Dengan struktur ini, sistem dapat mengolah data produksi dari berbagai wilayah secara lebih terorganisir dan menampilkan hasil analisis potensi komoditas perikanan secara efektif kepada pengguna.

2.1.14 Mysql

MySQL merupakan sebuah sistem manajemen basis data relasional (*Relational Database Management System / RDBMS*) berbasis sumber terbuka yang menggunakan *Structured Query Language (SQL)* sebagai bahasa standar utamanya. Sistem basis data ini secara fungsional mendistribusikan entitas data ke dalam tabel-tabel yang saling terhubung, sehingga memfasilitasi pengolahan informasi yang kompleks secara efisien dan terstruktur (Raihani, 2025).

Dalam pengembangan perangkat lunak berbasis *website*, MySQL memiliki tingkat kompatibilitas yang sangat tinggi ketika diintegrasikan dengan bahasa pemrograman skrip seperti PHP. Keunggulan komparatifnya terletak pada kemampuan skalabilitas dalam menangani volume data yang besar, kecepatan eksekusi kueri (*query*), serta fleksibilitas skema yang mendukung stabilitas sistem secara keseluruhan tanpa membebani kinerja peladen (Santoso, 2025).

Penggunaan MySQL pada penelitian ini berfungsi sebagai media penyimpanan utama (database) untuk sistem penentuan komoditas perikanan tangkap perairan darat berbasis website pada Dinas Kelautan dan Perikanan Provinsi Sumatera Selatan. Basis data ini



dirancang untuk menyimpan berbagai informasi yang diperlukan dalam sistem, seperti data produksi perikanan dari masing-masing wilayah, data jenis komoditas ikan, serta nilai kriteria yang digunakan dalam proses analisis menggunakan metode Rule-Based System. Dengan pengelolaan basis data yang terstruktur, sistem dapat melakukan proses pengolahan dan pengambilan informasi secara lebih cepat, akurat, dan terorganisir.

2.1.15 Laragon

Laragon merupakan perangkat lunak lingkungan pengembangan web lokal (*local web environment*) yang dirancang untuk menyederhanakan proses perancangan dan pengujian aplikasi berbasis web. Platform ini menyediakan ekosistem peladen terintegrasi yang bersifat ringan, cepat, dan portabel, memungkinkan pengembang untuk menjalankan simulasi arsitektur web secara penuh pada komputer pribadi tanpa memerlukan konfigurasi jaringan yang rumit (Anggraeni et al., 2026).

Secara teknis, perangkat lunak ini telah dilengkapi dengan bundel program pendukung esensial seperti Apache, PHP, dan MySQL yang terkonfigurasi secara otomatis. Fitur unggulan dari lingkungan peladen ini adalah kemampuannya dalam mengisolasi basis kode (*codebase*) serta mendukung manajemen basis data antarmuka secara langsung, sehingga siklus pemeliharaan aplikasi (*maintenance*) pada fase prapeluncuran menjadi jauh lebih stabil (Susanti et al., 2024).

Pemanfaatan platform Laragon pada penelitian ini digunakan sebagai lingkungan peladen lokal (*localhost*) dalam proses pengembangan dan pengujian aplikasi penentuan komoditas perikanan tangkap perairan darat berbasis website sebelum sistem diimplementasikan pada lingkungan server yang sebenarnya. Lingkungan pengembangan ini memungkinkan peneliti untuk melakukan pengujian interaksi antara antarmuka sistem, logika program yang dibangun menggunakan PHP dan Laravel, serta basis data MySQL. Dengan demikian, proses pengujian sistem dapat dilakukan secara optimal untuk memastikan bahwa penerapan metode Rule-Based System dalam menentukan potensi komoditas perikanan dapat berjalan dengan baik.



2.1.16 Visual Studio Code

Visual Studio Code adalah perangkat lunak penyunting teks sumber (*source code editor*) lintas platform yang dikembangkan secara sumber terbuka dan dirancang untuk memberikan lingkungan pengembangan terintegrasi (IDE) dengan performa tinggi. Perangkat lunak ini mendukung berbagai ragam sintaksis bahasa pemrograman modern dan difungsikan secara sentral untuk menulis, menyunting, serta mendokumentasikan struktur logika instruksi sebuah program secara komprehensif (Nur Aeni Hidayah, 2024).

Secara operasional, penyunting kode ini dilengkapi dengan ekosistem ekstensi yang sangat luas dan fitur analitik terapan seperti saran penulisan sintaks otomatis (*IntelliSense*). Fungsionalitas tersebut secara signifikan memfasilitasi pengembang perangkat lunak dalam melakukan proses penelusuran kesalahan logika (*debugging*) secara seketika di dalam editor, mengurangi anomali pengetikan struktural, serta menjaga konsistensi arsitektur kode pada sistem berskala besar (Ani Murani et al., 2025).

Pada penelitian ini, Visual Studio Code digunakan sebagai lingkungan kerja utama dalam proses pengembangan sistem penentuan komoditas perikanan tangkap



perairan darat berbasis website. Perangkat lunak ini dimanfaatkan untuk menulis, mengedit, serta mengelola struktur kode program yang dibangun menggunakan bahasa pemrograman PHP dan kerangka kerja Laravel. Melalui editor ini, peneliti mengimplementasikan logika metode s ke dalam bentuk kode program yang berfungsi untuk menganalisis data produksi perikanan dan menghasilkan informasi mengenai komoditas perikanan yang memiliki potensi unggul di wilayah perairan darat Provinsi Sumatera Selatan.

2.1.17 Penelitian Terdahulu

Berikut uraian penelitian terdahulu dapat dilihat pada Tabel 2.5.

Tabel 2.5 Penelitian Terdahulu

No	Nama (Tahun)	Judul	Metode	Hasil Penelitian
1.	(Mubarok & Romli, 2024)	Implementasi Metode <i>Rule Based</i> dalam Mendeteksi Serangan <i>Brute Force</i> pada <i>OwncLOUD</i>	<i>Rule-Based Method, Intrusion Detection System</i>	Penelitian ini mengimplementasikan metode <i>rule-based</i> untuk mendeteksi serangan <i>brute force</i> pada sistem <i>cloud storage OwncLOUD</i> . Sistem menganalisis pola <i>login</i> yang mencurigakan berdasarkan aturan yang telah ditentukan sehingga mampu membantu administrator dalam mendeteksi serangan keamanan jaringan secara lebih cepat dan efektif.
2.	(Husna et al.,	Aplikasi	<i>Rule – Based</i>	Penelitian ini



	2024)	Diagnosa Penyakit Tanaman Buah Menggunakan Metode <i>Rule Based Reasoning</i>	<i>Reasoning</i>	mengembangkan aplikasi berbasis web untuk mendiagnosa penyakit tanaman buah menggunakan metode <i>Rule-Based Reasoning</i> . Sistem bekerja dengan mencocokkan gejala yang dimasukkan pengguna dengan aturan dalam basis pengetahuan sehingga menghasilkan diagnosis penyakit serta rekomendasi penanganan yang tepat.
3.	(Alfian et al., 2024)	Sistem Pakar Penyakit Sapi Menggunakan <i>Rule Based Reasoning</i> dengan <i>Forward Chaining</i>	<i>Rule – Based Reasoning, Forward Chaining</i>	Penelitian ini merancang sistem pakar untuk mendiagnosa penyakit pada sapi dengan memanfaatkan metode <i>rule-based reasoning</i> dan <i>forward chaining</i> . Sistem mampu membantu peternak dalam mengidentifikasi penyakit sapi berdasarkan gejala yang muncul serta memberikan rekomendasi penanganan penyakit.
4.	(Fakhrizal & Kurniawan, 2024)	Sistem Informasi Pelayanan	<i>Rule-Based Decision Support</i>	Penelitian ini mengembangkan sistem pendukung keputusan



		Orang Tua Menggunakan <i>Rule-Based Decision Support System</i> untuk Meningkatkan Transparansi dan Komunikasi Antara Sekolah dengan Orang Tua	<i>System</i>	berbasis <i>rule-based</i> untuk meningkatkan transparansi komunikasi antara sekolah dan orang tua siswa. Sistem mengolah data nilai, kehadiran, dan pembayaran sekolah untuk memberikan informasi yang lebih akurat kepada orang tua siswa.
5.	(Fadila & Tanamal, 2021)	Penerapan <i>Rule-Based Expert System</i> dalam Perancangan Aplikasi Sistem Pakar untuk Mendiagnosa Penyakit Infeksi Saluran Pernapasan Akut Berbasis Android	<i>Rule – Based Expert System</i>	Penelitian ini mengembangkan aplikasi sistem pakar berbasis Android yang menggunakan metode <i>Rule-Based Expert System</i> untuk mendiagnosa penyakit ISPA berdasarkan gejala yang dialami pasien. Sistem memberikan hasil diagnosis dan saran penanganan penyakit.
6.	(Hermawan & Fachrudin, 2023)	Rancang Bangun Sistem Irigasi Cerdas Menggunakan Metode <i>Fuzzy</i>	<i>Fuzzy Rule-Based, IOT</i>	Penelitian ini merancang sistem irigasi cerdas berbasis IoT yang menggunakan metode <i>fuzzy rule-based</i> untuk mengatur pintu air



		<i>Rule-Based</i> untuk Otomatisasi Pintu Air dan Pendeteksian Endapan		secara otomatis. Sistem mampu mengontrol distribusi air secara efisien serta mendeteksi adanya endapan pada saluran irigasi.
7.	(Firdyanto & Rushendra, 2024)	<i>Implementation of Intrusion Detection System with Rule-Based Method on Website</i>	<i>Rule-Based Method, Intrusion Detection System</i>	Penelitian ini mengembangkan sistem deteksi intrusi berbasis website dengan menggunakan metode <i>rule-based</i> untuk mendeteksi aktivitas serangan pada jaringan. Sistem mampu mengenali pola serangan berdasarkan aturan yang telah ditentukan sebelumnya.
8.	(Putra et al., 2024)	Sistem Proses Validasi Pada Pembinaan Karir Prajurit Militer Dengan Metode <i>Rule-Based Reasoning</i> Berbasis <i>User Centered Design</i>	<i>Rule-Based Reasoning</i>	Penelitian ini membangun sistem <i>berbasis rule-based reasoning</i> untuk membantu proses validasi data dalam pembinaan karir prajurit militer. Sistem mampu meningkatkan akurasi data dan mengurangi kesalahan dalam proses administrasi.
9.	(Setiawan et	Implementasi	<i>Rule -Based</i>	Penelitian ini



	al., 2023)	Sistem Keamanan Jaringan Menggunakan <i>Rule-Based</i> IDS pada PT Netkrida Tuah Cakrawala	<i>Intrusion Detection System</i>	mengimplementasikan sistem keamanan jaringan berbasis <i>rule-based</i> IDS untuk mendeteksi aktivitas mencurigakan dalam jaringan perusahaan. Sistem mampu meningkatkan keamanan jaringan dengan memanfaatkan aturan deteksi serangan yang telah ditentukan.
10.	(Noorhadi & Mulyati, 2024)	Implementasi <i>Rule-Based Reasoning</i> dalam Pencegahan Stunting (Studi Kasus: Dinas Kesehatan Kabupaten Klaten)	<i>Rule-Based Reasoning</i>	Penelitian ini mengembangkan sistem berbasis <i>rule-based reasoning</i> untuk membantu identifikasi risiko stunting berdasarkan data status gizi masyarakat sehingga dapat mendukung pengambilan keputusan dalam program pencegahan stunting.