

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terkait Pengembangan *Backend* pada Sistem Aplikasi berbasis Website telah banyak dilakukan sebelumnya. Oleh karena itu, diperlukan kajian literatur untuk menghindari duplikasi penelitian. Dalam proyek akhir ini, terdapat sejumlah referensi dari penelitian sebelumnya yang dijadikan acuan dalam penyusunan proyek akhir ini, di antaranya adalah:

Penelitian pertama yang dilakukan oleh Azhari dkk (2025) merancang Sistem Informasi Manajemen Petugas Penyuluh untuk Dinas Pertanian Kabupaten Jombang guna mengatasi kendala pelaporan dan pengelolaan data manual yang menyulitkan monitoring kinerja penyuluhan. Sistem ini dikembangkan menggunakan metode waterfall, mencakup tahap analisis kebutuhan, perancangan sistem berbasis UML, implementasi dengan PHP dan *CodeIgniter*, serta pengujian *black box* yang valid. Hasilnya menunjukkan bahwa sistem mampu menyediakan data real-time yang memudahkan petugas dan dinas dalam tugas administrasi, evaluasi, dan pengambilan keputusan, serta mendukung peningkatan produktivitas pertanian secara berkelanjutan. Keberhasilan sistem ini menegaskan pentingnya integrasi teknologi informasi dalam layanan penyuluhan, sebagaimana ditegaskan oleh penelitian sebelumnya dan membuka peluang pengembangan lebih lanjut melalui integrasi kecerdasan buatan, peningkatan antarmuka pengguna, serta evaluasi sistem yang berkelanjutan untuk menjaga keandalan dan efektivitas layanan pertanian daerah. (Lestari, 2023)

Penelitian kedua oleh Rizka Indah dkk dari Universitas Bina Sarana Informatika ini merancang Sistem Informasi Penggajian berbasis web menggunakan framework Laravel dan metode *Agile Software Development* untuk menjawab tantangan efisiensi, akurasi, dan kecepatan dalam pengelolaan data gaji karyawan. Sistem ini dibangun dengan pendekatan iteratif yang mencakup tahapan perencanaan, implementasi, pengujian black-box, dokumentasi, deployment, dan pemeliharaan. Laravel dipilih karena sifatnya yang open source dan fitur modern seperti Eloquent ORM, Blade Template, dan Artisan CLI yang mempercepat proses pengembangan. Penggunaan Entity Relationship Diagram (ERD) dan Logical Record Structure

(LRS) memperjelas struktur data dan hubungan antar entitas dalam sistem, memungkinkan integrasi antara modul absensi, jabatan, akun, hingga slip gaji. Penelitian ini memperkuat temuan sebelumnya seperti oleh (Zulfian dkk,2021) yang menekankan pentingnya sistem terkomputerisasi untuk menghindari kesalahan dalam perhitungan gaji, serta didukung oleh pemanfaatan Agile yang adaptif terhadap perubahan kebutuhan pengguna sebagaimana disebutkan dalam studi (Rifqi dkk, 2025)

Penelitian ketiga ini merancang dan membangun sistem informasi company profile berbasis web pada PT. Hasna Satya Negara, sebuah perusahaan pengembang perumahan, dengan menggunakan metode Agile pendekatan Scrum untuk mengatasi kendala penyebaran informasi secara manual yang dinilai tidak efisien dan berbiaya tinggi. Sistem yang dikembangkan bertujuan untuk memberikan informasi produk secara digital dan terstruktur kepada calon pelanggan melalui berbagai fitur seperti pengelolaan berita, galeri, testimoni, brosur, dan langganan email, yang semuanya diatur dalam kerangka kerja *Scrum* melalui tahapan *product backlog*, *sprint backlog*, *sprint*, serta review dan *retrospective*. Tahapan ini memungkinkan pengembangan yang fleksibel dan adaptif terhadap perubahan kebutuhan pengguna, mempercepat proses implementasi, serta meningkatkan kolaborasi antar tim pengembang. Hasil akhir menunjukkan sistem mampu memenuhi kebutuhan fungsional dan non-fungsional, seperti kemudahan akses informasi dan tampilan antarmuka yang atraktif, yang pada akhirnya diharapkan dapat meningkatkan jangkauan promosi dan efisiensi operasional perusahaan (Nugraha dkk, 2021)

Penelitian keempat yang dilakukan oleh Saofikh Bagus Farchani dkk dari Universitas Amikom Purwokerto ini merancang dan mengimplementasikan sistem *backend* inventaris peminjaman berbasis REST API menggunakan Node.js untuk Balai Desa Widarapayung Kulon. Sistem ini dikembangkan dengan metode Waterfall, melalui tahapan analisis kebutuhan, desain sistem, implementasi, pengujian, deployment, dan pemeliharaan. Fokus penelitian terletak pada efisiensi komunikasi antara frontend dan *backend*, dengan implementasi fitur peminjaman, autentikasi JWT, serta pengelolaan pengguna dan history peminjaman. Node.js

dipilih karena kemampuannya mengelola permintaan asynchronous secara efisien, sementara pengujian

menggunakan metode Black Box memastikan setiap endpoint berfungsi sesuai spesifikasi. Penelitian ini menunjukkan bahwa sistem mampu meningkatkan transparansi dan efisiensi dalam pengelolaan aset publik, serta menambahkan nilai lebih dibandingkan penelitian sebelumnya yang belum mengintegrasikan autentikasi token dan pemisahan hak akses admin-user secara ketat (Akhdan dkk, 2023), membuktikan bahwa integrasi REST API berbasis Node.js efektif dalam membangun sistem inventaris modern yang responsif dan adaptif

Penelitian kelima yang dilakukan oleh Bayu Wijaya Putra dkk ini bertujuan untuk merancang Sistem Informasi Kelurahan Plaju Darat berbasis Agile Framework dengan pendekatan Scrum, guna mengatasi permasalahan pencatatan manual yang sering menimbulkan duplikasi data dan keterlambatan layanan administrasi. Penelitian ini menggunakan pendekatan kualitatif dan dikembangkan dalam lima sprint selama lima minggu, mencakup fitur login, pengelolaan data warga, potensi wilayah, inventaris barang, hingga arsip surat masuk. Sistem diuji menggunakan metode Black Box dan System Usability Scale (SUS), dengan hasil rata-rata 81,06 yang dikategorikan sebagai tingkat usability “Excellent”, menunjukkan penerimaan tinggi dari pengguna akhir. Peran Scrum dalam proses iteratif memungkinkan kolaborasi efektif antara tim pengembang dan pemangku kepentingan, serta memberikan ruang untuk adaptasi terhadap perubahan kebutuhan pengguna secara cepat, sejalan dengan temuan sebelumnya oleh Alexandes dkk, (2022) yang juga menunjukkan keberhasilan Scrum dalam proyek pengembangan sistem pelayanan publik.

Tabel 2.1 Perbandingan Penelitian Terdahulu

Nama Peneliti	(Azhari dkk, 2025)	(Rizka Indah dkk, 2025)	(Nugraha dkk, 2021)	(Bayu dkk, 2024)	(Saofikh dkk, 2023)
----------------------	--------------------	-------------------------	---------------------	------------------	---------------------

JUDUL	Sistem Informasi Manajemen Petugas Penyuluh untuk Dinas Pertanian Kabupaten Jombang	Sistem Informasi Penggajian Berbasis Web Menggunakan Laravel dan Agile Software Development	Sistem Informasi Company Profile Berbasis Web pada PT. Hasna Satya Negara	Sistem Informasi Kelurahan Plaju Darat	Perancangan dan Implementasi <i>Backend</i> Sistem Inventaris Peminjaman Berbasis REST API Menggunakan Node.js
METODE	Waterfall	Agile Software Development	Agile (Scrum)	Agile (Scrum), Kualitatif	Waterfall
FITUR	Manajemen data penyuluh, laporan penyuluh, dashboard evaluasi	Modul absensi, jabatan, penggajian, slip gaji, akun user	Pengelolaan berita, galeri, testimoni, brosur, langganan email	Login, data warga, potensi wilayah, inventaris barang, arsip surat	Peminjaman, autentikasi JWT, manajemen user, riwayat peminjaman
HASIL	Penyediaan data real-time untuk mendukung evaluasi, keputusan, dan peningkatan kinerja pertanian.	Sistem gaji efisien dan akurat, adaptif terhadap kebutuhan pengguna	Promosi digital efisien, informasi produk mudah diakses.	Usability “Excellent” (81,06), adaptif dan kolaboratif terhadap kebutuhan kelurahan	Komunikasi efisien frontend-backend, transparansi dan keamanan data meningkat

Penelitian saat ini menggunakan metode XP dan *framework laravel* untuk membangun sistem *backend* yang mempermudah pelaporan, *backup data*, dan

pemantauan kegiatan konservasi serta menyajikan edukasi informasi kepada masyarakat secara efektif tentang konservasi hutan. Pada penelitian ini adanya penyebaran informasi edukatif untuk masyarakat sekitar kawasan hutan konservasi. Informasi ini berupa jumlah pohon yang akan ditanam, jenis-jenis pohon yang digunakan, manfaat dari masing-masing jenis pohon, serta lokasi dan waktu kegiatan konservasi yang dapat tersampaikan secara merata dan mudah diakses oleh masyarakat. Dibandingkan penelitian sebelumnya yang fokus pada sistem penggajian, company profile, atau manajemen internal, penelitian ini lebih unggul karena bersifat kolaboratif antara instansi dan masyarakat, mendukung transparansi data, serta fleksibel terhadap perubahan kebutuhan lapangan melalui integrasi REST API dan pendekatan XP.

2.2 Landasan Teori

2.2.1. UPT-KPH Tasik Besar Serkap

UPT Kesatuan Pengelolaan Hutan (KPH) Tasik Besar Serkap Merupakan Unit Pelaksanaan teknis yang berada dibawah naungan UPT Dinas Lingkungan Hidup dan Kehutanan (DLHK) Provinsi Riau. UPT ini bertanggung jawab atas pengelolaan dan konservasi sumber daya hutan secara berkelanjutan di wilayah yang mencakup Kabupaten Siak, Indragiri Hilir, Pelalawan, serta sebagian wilayah Kota Pekanbaru. Dipimpin oleh Bapak Andri, S.Hut., UPT KPH Tasik Besar Serkap menjalankan berbagai program pelestarian lingkungan hidup, rehabilitasi hutan, serta pengawasan terhadap aktivitas kehutanan guna menjaga ekosistem tetap lestari dan bermanfaat bagi masyarakat sekitar.



Gambar 1. Kantor UPT KPH Tasik Besar Serkap

Sebagai salah satu KPH strategis di Provinsi Riau, UPT KPH Tasik Besar Serkap berperan penting dalam mengimplementasikan kebijakan pemerintah daerah terkait konservasi hutan dan pengelolaan lingkungan hidup. Wilayah kerjanya yang mencakup berbagai kabupaten dengan potensi hutan yang luas menjadikan UPT ini memiliki tanggung jawab besar dalam menekan laju deforestasi, mendukung upaya mitigasi perubahan iklim, serta memberdayakan masyarakat lokal melalui pemanfaatan hasil hutan non-kayu secara lestari. Selain itu, UPT ini juga aktif dalam kegiatan monitoring kawasan hutan, pengendalian kebakaran hutan dan lahan, serta pengembangan kawasan konservasi yang mendukung keanekaragaman hayati.

Dalam menjalankan tugasnya, UPT KPH Tasik Besar Serkap juga menjalin kerja sama dengan berbagai pihak, termasuk lembaga swadaya masyarakat, instansi pemerintah lainnya, serta masyarakat adat dan lokal. Kolaborasi ini bertujuan untuk meningkatkan efektivitas pengelolaan kawasan hutan, mendorong partisipasi aktif masyarakat dalam pelestarian lingkungan, serta menciptakan keseimbangan antara aspek ekologi, sosial, dan ekonomi. Adapun kantor UPT KPH Tasik Besar Serkap beralamat di Jl. Pengayoman No.1, Tengkerang Utara, Kecamatan Bukit Raya,

Kota Pekanbaru, Riau 28126, yang juga menjadi pusat koordinasi kegiatan dan layanan informasi kepada masyarakat.

2.2.2. Konservasi Hutan

Konservasi hutan adalah upaya perlindungan, pengelolaan, dan pemanfaatan sumber daya hutan secara berkelanjutan agar fungsi ekologis, ekonomis, dan sosialnya tetap terjaga. Hutan berperan penting sebagai habitat bagi lebih dari 80% spesies di dunia, sehingga kerusakan atau fragmentasi hutan berdampak langsung pada hilangnya keanekaragaman hayati. Dalam perspektif biologi konservasi, tujuan utama konservasi hutan adalah menjaga kelestarian flora, fauna, dan ekosistemnya, mempertahankan variasi genetik, serta memastikan proses ekologis tetap berjalan secara alami.



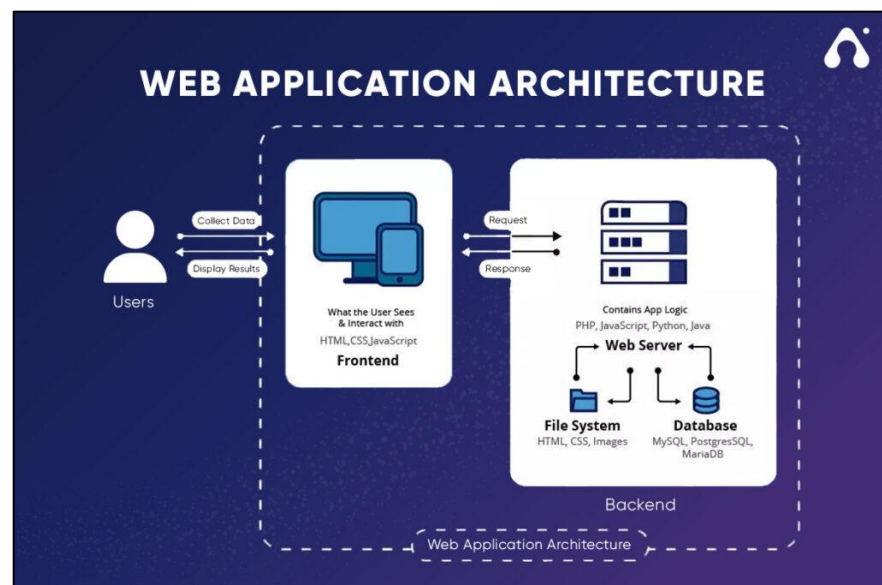
Gambar 2 1. Kegiatan Konservasi

Ancaman terhadap hutan di Indonesia meliputi eksploitasi berlebihan melalui penebangan liar dan pembakaran hutan, fragmentasi habitat yang mengurangi luas interior hutan, polusi yang merusak tanah dan air, serta invasi spesies asing yang menggeser vegetasi asli. Perubahan iklim juga memperburuk kondisi ekosistem hutan dengan mengubah pola curah hujan, suhu, dan kelembapan yang pada akhirnya memengaruhi struktur vegetasi dan keberlangsungan satwa liar. Jika tidak dikelola dengan baik, kerusakan hutan akan mengakibatkan hilangnya jasa lingkungan penting seperti pengaturan tata air, perlindungan tanah, dan penyerapan karbon.

Strategi konservasi hutan dilakukan melalui dua pendekatan utama, yaitu *in-situ* dan *ex-situ*. Konservasi *in-situ* dilakukan di habitat asli melalui penetapan kawasan lindung seperti taman nasional, cagar alam, suaka margasatwa, dan hutan lindung, yang bertujuan mempertahankan ekosistem secara utuh. Sementara itu, konservasi *ex-situ* dilakukan di luar habitat alami, misalnya melalui penangkaran satwa, kebun botani, dan bank gen, terutama untuk spesies yang populasinya di alam sudah kritis. Prinsip konservasi hutan menekankan pada perlindungan keanekaragaman hayati, pencegahan kepunahan cepat, pemeliharaan kompleksitas ekologi, dan dukungan terhadap proses evolusi alami agar fungsi hutan tetap lestari bagi generasi sekarang dan mendatang (Utami I, 2022).

2.2.3. Pengertian *Backend*

Backend merupakan komponen penting dalam arsitektur sistem informasi yang bekerja di balik layar (*server-side*). Komponen ini bertanggung jawab dalam menjalankan logika bisnis, memproses permintaan pengguna, dan mengelola basis data. Tidak seperti frontend yang berinteraksi langsung dengan pengguna, *backend* lebih fokus pada bagaimana data disimpan, diproses, dan dikirim kembali ke antarmuka pengguna dengan aman dan optimal. Fungsi-fungsi seperti autentikasi pengguna, pengelolaan data, serta pemrosesan transaksi semuanya dijalankan oleh sistem *backend* (Tummers, 2021).



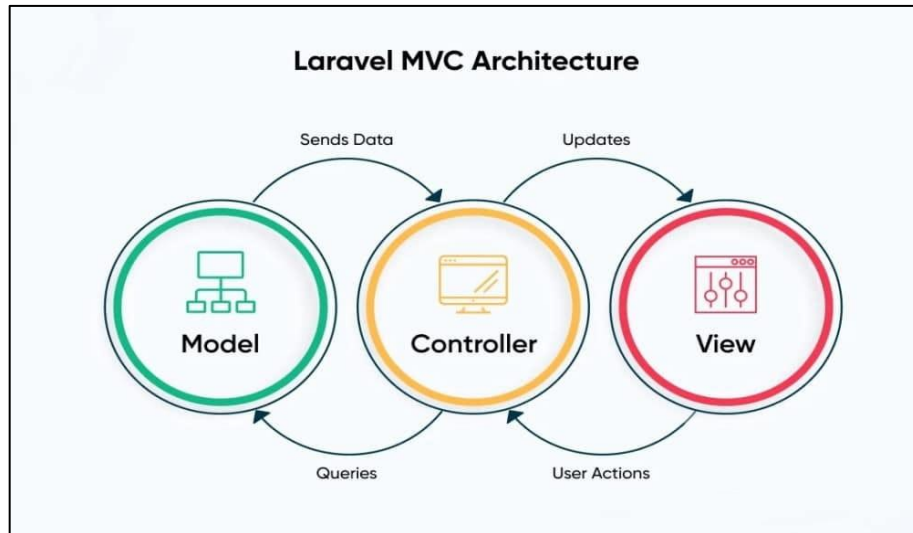
Gambar 2. *Web Application Architecture*

Menurut Sommerville (2016), *backend* menyediakan layanan utama bagi sistem perangkat lunak dengan menangani logika aplikasi dan proses data yang tidak terlihat oleh pengguna. *Backend* menjadi jembatan antara pengguna dan database, memastikan bahwa permintaan pengguna dapat diproses secara tepat dan sesuai dengan aturan bisnis yang telah ditentukan. Dalam konteks sistem informasi konservasi, *backend* bertugas untuk mencatat data kegiatan konservasi, status area hutan, serta aktivitas pengguna sistem secara terpusat dan terorganisir.

Selain itu, menurut Welling dan Thomson (2017), pengembangan *backend* umumnya dilakukan dengan bahasa pemrograman seperti *PHP*, *Python*, atau *Node.js*, dan memanfaatkan sistem manajemen basis data seperti *MySQL* atau *PostgreSQL*. Dalam proyek ini, *Laravel* dipilih sebagai framework backend karena menyediakan arsitektur yang terstruktur, aman, dan efisien, serta mendukung pengembangan REST API secara optimal. Dengan peran strategis ini, backend menjadi komponen inti yang memastikan integritas data, keamanan sistem, serta keberlanjutan layanan dalam pengelolaan informasi konservasi hutan secara digital (Mubariz dkk, 2020).

2.2.4. Framework Laravel

Laravel merupakan salah satu *framework* PHP modern yang bersifat open-source dan dirancang untuk mempermudah serta mempercepat proses pengembangan aplikasi web. *Laravel* dibangun menggunakan arsitektur *Model-View-Controller* (MVC) yang memisahkan logika aplikasi, tampilan, dan pengelolaan data sehingga sistem lebih terstruktur dan mudah dikembangkan. Framework ini sangat populer di kalangan pengembang karena menyederhanakan banyak aspek pengembangan backend seperti routing, autentikasi, pengiriman email, hingga manajemen database.



Gambar 3. *MVC Laravel*

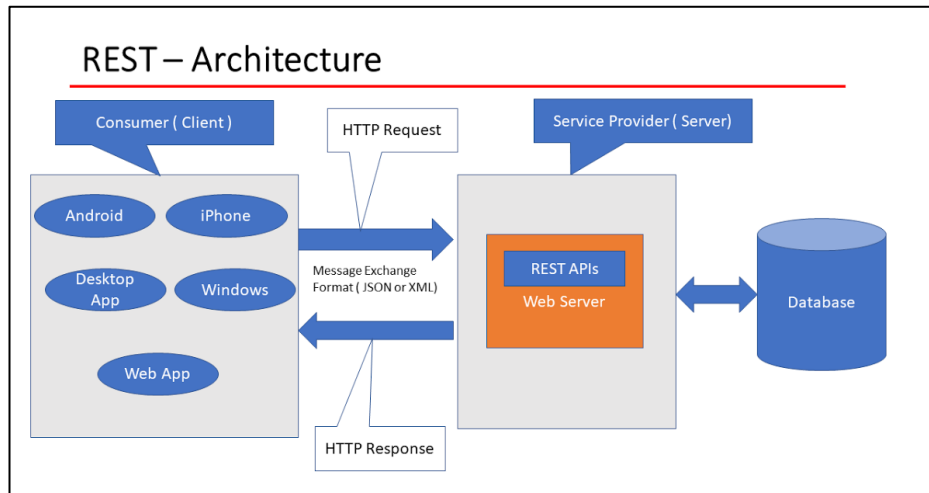
Menurut Otwell (2020), pencipta Laravel, framework ini dirancang untuk menghadirkan sintaks yang ekspresif dan elegan, sehingga memudahkan developer untuk menulis kode yang bersih dan mudah dipelihara. Laravel menyediakan fitur-fitur unggulan seperti Eloquent ORM untuk interaksi dengan database, Blade sebagai engine template bawaan, serta Laravel Artisan yang merupakan command line interface untuk mempermudah manajemen aplikasi. Selain itu, Laravel mendukung pembuatan RESTful API secara native, yang sangat penting dalam pengembangan aplikasi berbasis layanan data seperti sistem informasi konservasi.

Fitur keamanan yang tersedia di Laravel juga menjadi alasan mengapa framework ini banyak digunakan. Laravel memiliki proteksi terhadap serangan umum seperti *SQL Injection*, *Cross Site Scripting (XSS)*, dan *Cross Site Request Forgery (CSRF)*. Selain itu, Laravel menyediakan sistem autentikasi dan otorisasi pengguna yang dapat disesuaikan dengan kebutuhan aplikasi. Dengan adanya sistem middleware, pengembang dapat dengan mudah mengelola akses ke berbagai bagian aplikasi sesuai dengan peran pengguna, baik admin pusat maupun admin lapangan seperti pada proyek ini (Sunardi & Suharjito, 2019).

2.2.5. REST API

REST API (Representational State Transfer Application Programming Interface) adalah sebuah arsitektur layanan web yang memungkinkan sistem backend untuk berkomunikasi dengan frontend atau aplikasi lain secara efisien menggunakan protokol HTTP. Konsep REST pertama kali diperkenalkan oleh Roy

Fielding pada tahun 2000 dalam disertasinya, dengan tujuan menciptakan sistem komunikasi yang sederhana, ringan, dan dapat diskalakan. REST API banyak digunakan dalam pengembangan aplikasi web modern karena dapat diakses melalui berbagai platform seperti web, mobile, hingga perangkat IoT.



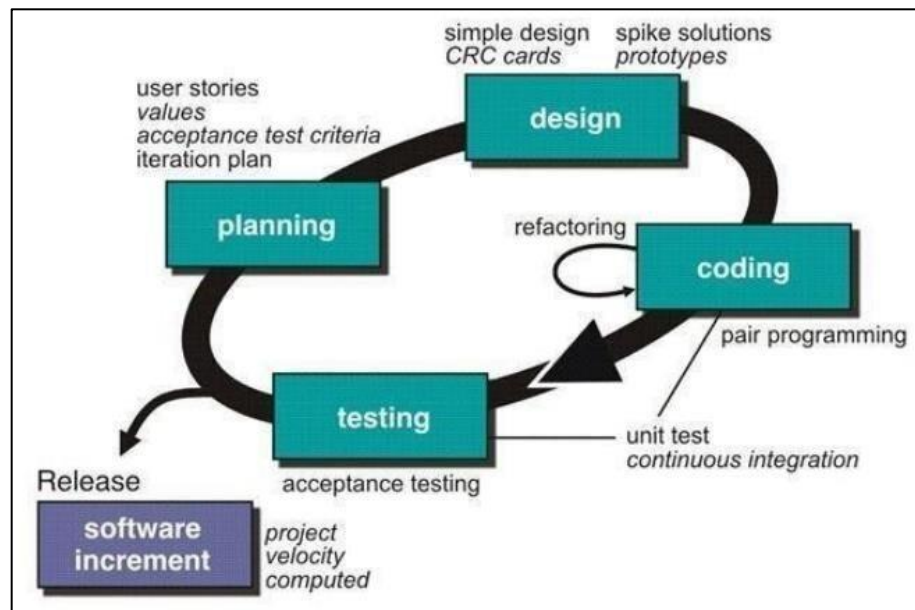
Gambar 4. *RESTful API*

Dalam sistem berbasis REST, setiap sumber daya (*resource*) seperti data pengguna, kegiatan konservasi, atau data lokasi, direpresentasikan sebagai URL yang dapat diakses menggunakan metode HTTP standar seperti *GET* (mengambil data), *POST* (menambah data), *PUT/PATCH* (memperbarui data), dan *DELETE* (menghapus data). Menurut Richardson & Ruby (2007), pendekatan ini mempermudah integrasi antara frontend dan *backend* karena masing-masing komponen hanya perlu memahami struktur permintaan dan respons data berbasis JSON atau XML. Hal ini sangat cocok untuk sistem informasi konservasi yang membutuhkan pertukaran data secara real-time dan fleksibel (Senduk dkk, 2023).

2.2.6. Metode *Extreme Programming* (XP)

Extreme Programming (XP) adalah salah satu metode dalam pengembangan perangkat lunak Agile yang berfokus pada peningkatan kualitas kode dan kemampuan adaptasi terhadap perubahan kebutuhan. Metode ini terdiri dari beberapa tahapan utama yang saling terhubung dalam siklus iteratif, yakni *planning*, *design*, *coding*, *testing*, dan *software increment (release)*. Pada tahap *planning*, tim menyusun *user stories* berdasarkan kebutuhan pengguna, menentukan nilai dan kriteria penerimaan, serta merencanakan iterasi. Pelanggan

terlibat langsung dalam proses ini untuk menentukan prioritas pengembangan, sehingga hasil akhir lebih sesuai kebutuhan (Septiani & Yanti, 2021) .



Gambar 5. Proses Alur XP

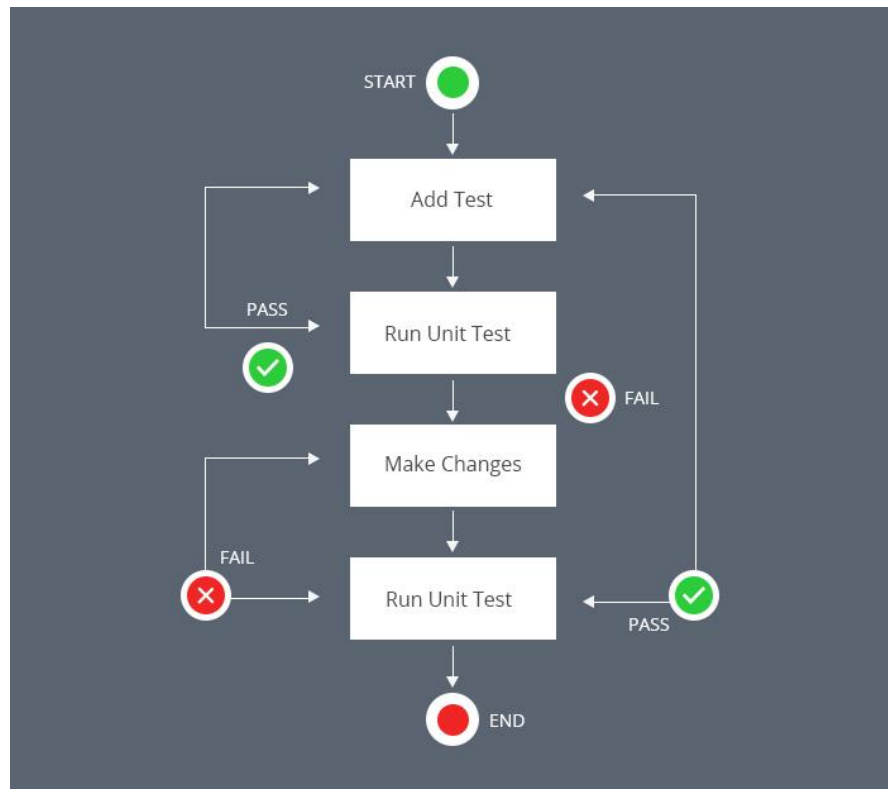
Selanjutnya, tahap design berfokus pada perancangan sederhana yang mengacu pada kebutuhan pengguna. Dalam tahap ini, digunakan alat bantu seperti *CRC Card* untuk mendefinisikan tanggung jawab dan relasi antar kelas, serta *Spike Solution* untuk mengeksplorasi prototipe solusi teknis secara cepat. Pendekatan ini sejalan dengan metode *Extreme Programming* (XP) yang menekankan kesederhanaan desain, eksplorasi awal terhadap potensi masalah teknis, dan kesiapan sistem dalam menghadapi pengembangan lebih lanjut (Baswardono dkk, 2024). Prinsip desain sederhana menjadi acuan utama agar sistem tetap fleksibel dan mudah dikembangkan. Pada tahap coding, praktik seperti *pair programming* dan *refactoring* digunakan untuk menjaga kualitas dan konsistensi kode. Semua anggota tim memiliki kepemilikan kode secara kolektif, memungkinkan siapa saja untuk memperbaiki atau meningkatkan bagian mana pun dari sistem. Tahap testing merupakan bagian penting dalam XP, di mana pengujian dilakukan sejak awal menggunakan *unit test* dan *acceptance test* untuk memastikan setiap komponen berfungsi sesuai harapan. Proses *continuous integration* memungkinkan integrasi kode secara terus-menerus, sehingga kesalahan dapat terdeteksi lebih dini. Setelah pengujian berhasil, perangkat lunak dirilis dalam bentuk *software increment* yang

siap digunakan. Rilis dilakukan secara berkala dalam ukuran kecil dan cepat, memungkinkan pengguna memberikan umpan balik yang akan digunakan dalam iterasi selanjutnya. Pada tahap ini juga dihitung *project velocity* untuk mengukur kecepatan tim dalam menyelesaikan fitur. Seluruh proses dalam XP saling terhubung dalam lingkaran umpan balik berkelanjutan, menjadikan metode ini sangat cocok untuk pengembangan sistem yang kompleks dan dinamis.

XP menggunakan sejumlah praktik utama seperti pengembangan berbasis pengujian (*Test-Driven Development*), pemrograman berpasangan (*Pair Programming*), integrasi kontinu, serta refaktorisasi kode. Praktik-praktik ini membantu tim dalam menjaga kualitas kode dan menghindari kesalahan sejak awal proses pengembangan. Selain itu, XP juga menerapkan konsep seperti kepemilikan kode kolektif, metafora sistem, rilis kecil yang sering, dan keterlibatan langsung pelanggan dalam proses pengembangan. Semua praktik tersebut mendorong adanya iterasi cepat dan kolaboratif, sehingga mempermudah tim untuk beradaptasi dengan kebutuhan yang terus berubah (Runeson dkk, 2020).

2.2.7. Unit Testing

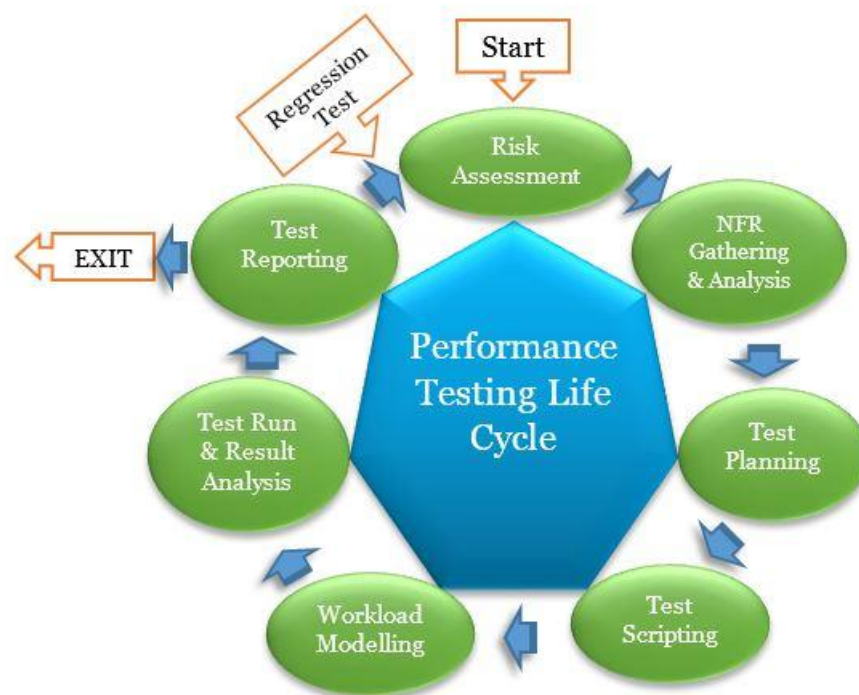
Unit Testing adalah proses pengujian perangkat lunak yang bertujuan untuk memverifikasi bahwa setiap fungsi atau unit terkecil dalam *backend* berjalan dengan benar secara independen. Dalam konteks pengembangan sistem informasi berbasis Laravel, unit testing dapat diterapkan pada fungsi dalam controller, model, atau service class yang bertanggung jawab atas logika bisnis tertentu. Dengan menggunakan tool seperti PHPUnit, pengembang dapat membuat skrip otomatis untuk menguji kondisi input dan output dari setiap fungsi secara detail. Keuntungan utama dari unit testing adalah kemampuannya dalam mendeteksi kesalahan sejak dini, sehingga memudahkan pemeliharaan kode dan mencegah error merambat ke bagian lain dari sistem (Jeremmy, 2025)



Gambar 6. Proses Alur *Unit Testing*

2.2.8. *Performance Testing*

Performance testing adalah jenis pengujian non-fungsional yang bertujuan untuk menentukan bagaimana kinerja suatu sistem dalam hal responsivitas dan stabilitas di bawah beban kerja tertentu. Tujuan utama dari pengujian ini adalah untuk mengidentifikasi hambatan kinerja serta memastikan bahwa perangkat lunak dapat memenuhi persyaratan kinerja yang telah ditetapkan. Pengujian ini sangat penting dilakukan sebelum sistem diluncurkan ke lingkungan produksi guna mencegah potensi kegagalan saat digunakan oleh pengguna nyata.



Gambar 7. Proses Alur *Performance Testing*

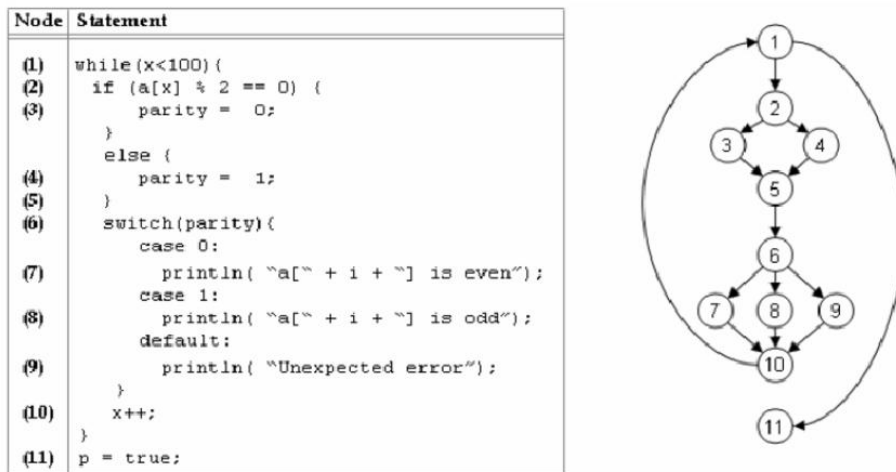
Terdapat beberapa jenis performance testing yang umum digunakan. **Load Testing** bertujuan untuk mengukur kinerja sistem saat beroperasi dalam beban kerja normal, seperti jumlah pengguna atau transaksi yang diharapkan. **Stress Testing** dilakukan untuk mengetahui batas maksimum sistem dengan cara meningkatkan beban hingga melampaui kapasitas normal. **Scalability Testing** mengevaluasi kemampuan sistem dalam menyesuaikan kinerjanya saat terjadi peningkatan atau penurunan beban. **Endurance Testing (Soak Testing)** fokus pada kestabilan sistem dalam jangka waktu panjang untuk mengidentifikasi masalah seperti kebocoran memori. Sedangkan **Spike Testing** menguji respon sistem terhadap lonjakan beban secara tiba-tiba, seperti saat banyak pengguna mengakses sistem secara bersamaan dalam waktu singkat (Pargaonkar, 2023).

2.2.9. *White Box Testing*

1. *Cyclomatic Complexity*

Cyclomatic Complexity adalah sebuah metrik dalam rekayasa perangkat lunak yang digunakan untuk mengukur kompleksitas logika dari sebuah program dengan

menghitung jumlah jalur independen yang dapat dijalankan dalam kode tersebut. Metrik ini diperkenalkan oleh Thomas J. McCabe pada tahun 1976 dan biasanya dihitung berdasarkan grafik alur kontrol (*Control Flow Graph*) dari program, di mana simpul (node) mewakili blok pernyataan dan sisi (*edge*) mewakili alur eksekusi antar blok tersebut. Rumus standar yang digunakan adalah $V(G) = E - N + 2P$, dengan E sebagai jumlah edge, N sebagai jumlah node, dan P sebagai jumlah komponen terhubung dalam program. Nilai *Cyclomatic Complexity* memberikan gambaran tentang seberapa rumit sebuah fungsi atau modul, dimana nilai yang tinggi menunjukkan kompleksitas yang besar dan potensi risiko kesalahan yang lebih tinggi, serta kebutuhan pengujian yang lebih intensif.



Gambar 8. Proses Alur *Cyclomatic Complexity*

Metrik ini sangat berguna dalam perencanaan pengujian white-box, khususnya untuk menentukan jumlah minimal kasus uji yang diperlukan agar semua jalur logika teruji dengan baik. Dalam beberapa tahun terakhir, berbagai penelitian telah mengembangkan variasi dari *Cyclomatic Complexity* seperti *Cross Cyclomatic Complexity* dan *Hybrid Cyclomatic Complexity* yang mengadaptasi metrik ini untuk lingkungan pemrograman modern dan model prediksi cacat perangkat lunak (Putri dkk, 2024).