

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian ini merujuk pada beberapa studi terdahulu yang berkaitan dengan sistem informasi sekolah sepak bola untuk memperoleh gambaran tentang kelebihan dan kekurangan dari sistem yang telah dibangun sebelumnya. Studi dari Wahyudi dkk. (2022) menunjukkan bahwa sistem informasi yang dikembangkan sudah mampu mengelola data pelatih dan siswa, namun belum menyediakan fitur evaluasi performa dan masih menggunakan antarmuka yang kurang modern. Selanjutnya, Putra dkk. (2021) mengembangkan sistem akademik berbasis *PHP dan MySQL* yang telah mendigitalisasi proses akademik, namun belum menyediakan fitur pelaporan prestasi siswa dan belum melibatkan orang tua secara aktif dalam proses pendidikan.

Penelitian oleh Amin (2020) menggunakan pendekatan *Agile* dalam mengembangkan sistem informasi sekolah olahraga berbasis *web* yang telah mendukung pelaporan nilai dan presensi kepada orang tua, tetapi belum menerapkan *framework* modern pada sisi antarmuka pengguna. Kemudian, Sari dan Nugroho (2021) merancang sistem informasi ekstrakurikuler dengan menggunakan *Laravel dan Vue.js* yang memiliki antarmuka menarik dan interaktif, namun belum menyediakan modul evaluasi performa siswa secara khusus.

Adapun penelitian saat ini difokuskan pada pengembangan *frontend* sistem informasi profil sekolah sepak bola berbasis *web* pada Sekolah Sepak Bola Rumbai Pratama. Sistem ini dirancang untuk mendukung promosi dan manajemen data siswa secara terpusat melalui antarmuka yang modern dan menarik. Namun, sistem ini masih berada pada tahap awal pengembangan, dan bagian *backend*-nya belum dibahas secara rinci. Hasil dari tinjauan ini menjadi dasar untuk mengembangkan sistem yang lebih komprehensif dan sesuai dengan kebutuhan pengguna, khususnya dalam aspek evaluasi performa, pelaporan prestasi, dan keterlibatan orang tua, di mana seluruh hasil penelitian

terdahulu tersebut telah dirangkum secara sistematis pada Tabel 2.1 berikut

Tabel 2.1 Penelitian Terdahulu

No	Peneliti & Tahun	Judul Penelitian dan Teknologi	Hasil
1	Wahyudi, D. R., dkk. (2022)	Perancangan Sistem Informasi Manajemen Sekolah Sepak Bola Mandala Majalengka menggunakan metode <i>waterfall</i>	Sistem dapat mengelola data pelatih dan siswa, namun belum mendukung evaluasi performa serta belum menggunakan desain antarmuka modern.
2	Putra, dkk (2021)	Perancangan Sistem Informasi Akademik Sekolah Sepak Bola Menggunakan <i>PHP dan MySQL</i>	Proses akademik sudah terdigitalisasi, tetapi belum menyediakan fitur pelaporan prestasi siswa dan keterlibatan orang tua masih minim.
3	Amin, M. (2020)	Pengembangan Sistem Informasi sekolah Olahraga Berbasis <i>Web</i> menggunakan metode pendekatan <i>Agile</i>	Sistem sudah mendukung pelaporan nilai dan presensi kepada orang tua, namun belum menerapkan <i>framework</i> modern pada tampilan antarmuka pengguna.
4	Sari, I., & Nugroho, H. (2021)	Rancang Bangun Sistem Informasi Ekstrakurikuler sekolah menggunakan pustaka <i>framework Laravel dan Vue.js</i>	Antarmuka pengguna menarik dengan <i>dashboard</i> interaktif, tetapi belum memiliki interaksi kuat antar penggunaannya.
5	Penelitian Saat Ini	Rancang Bangun <i>Backend</i> Sistem Informasi Profil Sekolah Sepak Bola Berbasis <i>Web</i> sebagai Media Promosi dan Manajemen Data Siswa (Studi Kasus: Sekolah Sepak Bola Rumbai Pratama)	Fokus pada pengembangan <i>frontend</i> yang modern, yang mendukung fitur promosi dan manajemen data siswa. Namun sistem masih dalam tahap awal dan pengembangan <i>backend</i> belum dijabarkan secara rinci.

Penelitian ini memiliki perbedaan dengan penelitian sebelumnya karena mengkaji permasalahan dengan menyesuaikan pendekatan penelitian terhadap kebutuhan dan karakteristik objek penelitian. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi dalam menjawab perumusan masalah

serta melengkapi hasil-hasil penelitian yang telah ada sebelumnya.

2.2 Landasan Teori

Perancangan dan implementasi *backend* sistem Sekolah Sepak Bola Rumbai Pratama didasarkan pada pemanfaatan *tools* pengembangan, penerapan metode pengujian, serta dukungan teori-teori yang relevan untuk menghasilkan sistem yang baik dalam mengelola proses manajemen data, penyimpanan data, dan mempromosikan sekolah sepak bola.

2.2.1 Sistem Informasi Sekolah Sepak Bola

Sistem informasi sekolah sepak bola merupakan aplikasi berbasis digital yang dirancang untuk mempermudah pengelolaan kegiatan di lingkungan SSB, seperti administrasi, komunikasi, evaluasi performa, dan dokumentasi. Sistem ini biasanya dapat diakses melalui perangkat *web* maupun *mobile*, sehingga memungkinkan berbagai pihak seperti pelatih, admin, dan orang tua untuk berinteraksi dan memperoleh informasi dengan lebih cepat dan efisien. Menurut Wahyudi et al. (2022), “Dengan adanya sistem ini, diharapkan efisiensi operasional dapat meningkat, dan transparansi informasi kepada orang tua serta pelatih dapat terjaga dengan baik”. Dengan adanya sistem yang terpusat, semua informasi penting seperti jadwal latihan, nilai evaluasi siswa, dan data pendaftaran dapat disimpan secara aman dan mudah diakses kapan saja. Oleh karena itu, penggunaan sistem informasi semacam ini akan sangat membantu dalam meningkatkan keteraturan dan transparansi dalam proses pembinaan, terutama dalam mendukung komunikasi yang lancar antara pihak sekolah sepak bola dengan orang tua siswa.

2.2.2 Laravel

Laravel merupakan salah satu *framework* berbasis PHP yang digunakan untuk membantu proses pengembangan aplikasi berbasis *web*. Laravel menyediakan berbagai *library* yang dapat digunakan dalam proses pengembangan sistem sehingga pengembang dapat membangun fungsi yang dibutuhkan dalam aplikasi dengan lebih terstruktur. Menurut Kadim dkk. (2023), *framework* Laravel dapat dimanfaatkan dalam pengembangan sistem berbasis web dengan menggunakan *library* yang tersedia pada Laravel.

Penggunaan Laravel dapat membantu proses pengembangan sistem karena menyediakan berbagai fitur yang mendukung pembuatan aplikasi *web*. Dalam penelitian Kadim dkk. (2023), Laravel digunakan untuk membangun sistem E-Notulen berbasis *web* mulai dari proses pengembangan hingga implementasi sistem. Oleh karena itu, Laravel dapat digunakan sebagai *framework* dalam pengembangan aplikasi berbasis *web* yang membutuhkan pengelolaan fungsi sistem secara terstruktur.

2.2.3 Metode Agile Scrum

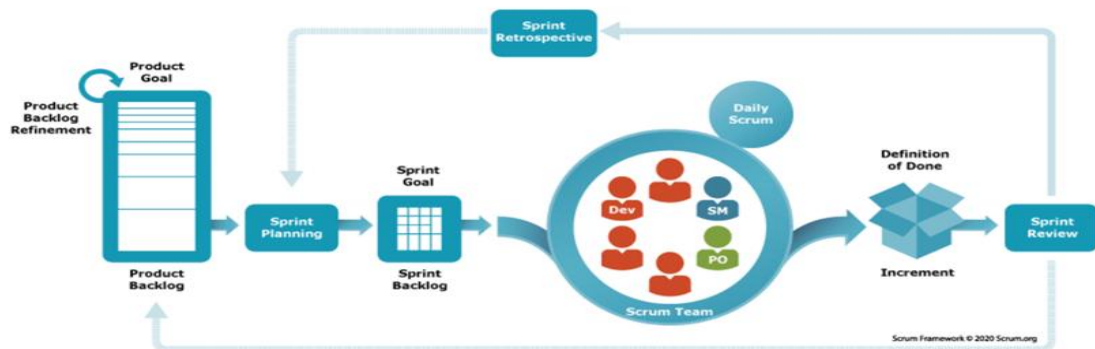
Metodologi Agile merupakan pendekatan dalam pengembangan perangkat lunak yang bersifat iteratif dan fleksibel terhadap perubahan kebutuhan selama proses pengembangan berlangsung. *Agile* membagi proses pengembangan menjadi siklus-siklus pendek atau disebut juga *iteration*, sehingga pengembang dapat memberikan hasil yang bisa langsung diuji dalam waktu singkat. Salah satu kerangka kerja yang paling umum digunakan dalam metodologi *Agile* adalah *Scrum*.

Scrum memungkinkan tim pengembang untuk bekerja secara kolaboratif dalam siklus kerja yang disebut *sprint*. Setiap *sprint* biasanya berlangsung antara dua hingga empat minggu, dengan tujuan menghasilkan *increment* atau bagian dari sistem yang siap digunakan. Menurut Schwaber dan Sutherland (2020), dalam pendekatan *Scrum*, proses pengembangan dilakukan melalui *Sprint* dengan durasi tetap selama satu bulan atau kurang. Dalam setiap *Sprint* terdapat beberapa kegiatan utama, yaitu *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*. Kegiatan tersebut dilakukan secara teratur untuk membantu tim mencapai tujuan *Sprint*, meninjau hasil pekerjaan, serta melakukan perbaikan pada proses kerja untuk *Sprint* berikutnya.

Struktur tim dalam *Scrum* terdiri dari tiga peran utama, yaitu *Product Owner* yang bertanggung jawab terhadap visi produk dan prioritas pengembangan, *Scrum Master* yang memfasilitasi proses kerja dan mengatasi hambatan, serta *Development Team* yang menjalankan proses pengembangan secara teknis. Selain

itu, *Scrum* juga memiliki tiga artefak utama, yakni *Product Backlog* sebagai daftar kebutuhan yang harus dipenuhi, *Sprint Backlog* sebagai rincian tugas dalam satu *sprint*, dan *Increment* sebagai hasil kerja yang telah selesai dan memenuhi kriteria penerimaan.

Berdasarkan kompleksitas kebutuhan yang ada di SSB Rumbai Pratama, seperti pengelolaan pendaftaran *online*, presensi siswa, evaluasi performa, hingga jadwal latihan, *Scrum* sangat cocok untuk diterapkan. Sistem yang dibangun dapat dikembangkan secara bertahap dan disesuaikan dengan masukan dari pengguna seperti pelatih, orang tua, dan admin. Pendekatan ini memungkinkan sistem terus berkembang seiring dengan kebutuhan di lapangan tanpa harus menunggu proyek selesai secara keseluruhan. Dengan menggunakan *Scrum*, pengembangan sistem informasi untuk SSB ini dapat berjalan lebih adaptif dan responsif terhadap perubahan yang mungkin muncul di tengah proses.



Gambar 2.1 *Scrum Agile Methodology*

2.2.4 *Software Development Life Cycle (SDLC)*

Software Development Life Cycle atau *SDLC* adalah sebuah kerangka kerja sistematis yang digunakan untuk mengembangkan perangkat lunak secara bertahap, mulai dari tahap perencanaan hingga pemeliharaan. *SDLC* membantu pengembang untuk merancang, membangun, dan memelihara perangkat lunak dengan lebih terstruktur dan cepat, serta meminimalkan risiko kesalahan selama proses pengembangan berlangsung. Menurut Nova et al. (2022), *Software Development Life Cycle (SDLC)* merupakan serangkaian aktivitas dalam pengembangan perangkat lunak yang meliputi proses mendefinisikan, mengembangkan, menguji,

mengirim, mengoperasikan, dan memelihara sistem. Salah satu metode yang dapat digunakan dalam *SDLC* adalah *Agile*, yang banyak diterapkan dalam pengembangan sistem informasi berbasis website.

Tahap pertama dalam *SDLC* adalah perencanaan, di mana dilakukan identifikasi kebutuhan sistem dan kelayakan proyek baik dari sisi teknis, ekonomi, maupun operasional. Selanjutnya adalah tahap analisis, di mana pengembang mencoba memahami lebih dalam masalah yang dihadapi pengguna serta merumuskan solusi melalui pemodelan sistem seperti *use case* dan diagram alur data. Setelah itu, masuk ke tahap desain, yaitu proses membuat rancangan teknis dari sistem, termasuk desain *database*, struktur antarmuka pengguna, serta arsitektur sistem secara keseluruhan.

Tahap implementasi dilakukan setelah desain disetujui, di mana pengembang mulai menulis kode program sesuai dengan desain yang telah dibuat. Kemudian sistem akan masuk ke tahap pengujian (*testing*), untuk memastikan semua fitur berjalan dengan baik dan tidak ditemukan kesalahan (*bug*) sebelum diserahkan kepada pengguna. Tahap terakhir adalah pemeliharaan (*maintenance*), yang dilakukan setelah sistem digunakan. Tahap ini mencakup perbaikan *bug*, peningkatan sistem, dan pembaruan berdasarkan umpan balik dari pengguna.

Dengan menggunakan pendekatan *SDLC*, proses pengembangan sistem informasi seperti sistem informasi SSB menjadi lebih terkontrol dan sistematis, serta mampu beradaptasi dengan perubahan kebutuhan yang mungkin terjadi.

2.2.5 Visual Studio Code (VS Code)

Visual Studio Code atau *VS Code* merupakan editor kode yang digunakan untuk membantu proses pengembangan perangkat lunak. *VS Code* dapat digunakan untuk menulis dan mengelola kode program serta mendukung berbagai bahasa pemrograman. Selain itu, *VS Code* memiliki berbagai fitur dan ekstensi yang dapat disesuaikan dengan kebutuhan pengembang dalam membuat aplikasi.

Menurut Hidayah dan Rofiqoh (2024), *Visual Studio Code* merupakan salah satu perangkat lunak yang digunakan dalam proses pengembangan program dan memiliki berbagai fitur untuk membantu pengguna dalam melakukan pengkodean.

Berdasarkan pengujian usability menggunakan metode Nielsen's Attributes of Usability (NAU), penggunaan Visual Studio Code dinilai berdasarkan beberapa aspek kemudahan yang berkaitan dengan pengalaman pengguna dalam menggunakan perangkat lunak tersebut.

Kemudahan penggunaan dan fitur yang tersedia pada *Visual Studio Code* dapat membantu pengembang dalam melakukan proses pengkodean dengan lebih mudah. Selain itu, dukungan terhadap berbagai bahasa pemrograman membuat *VS Code* dapat digunakan dalam berbagai kebutuhan pengembangan aplikasi. Pada penelitian ini, *Visual Studio Code* digunakan sebagai editor kode dalam proses pengembangan *backend* Sistem Informasi Sekolah Sepak Bola Rumbai Pratama.

2.2.6 XAMPP

XAMPP merupakan paket perangkat lunak yang menyediakan lingkungan *server* lokal untuk keperluan pengembangan aplikasi berbasis *web*. Paket ini mencakup *web server* Apache, *database* MySQL, serta dukungan bahasa pemrograman seperti PHP dan Perl yang umum digunakan dalam pengembangan sistem informasi. Dengan adanya *XAMPP*, pengembang dapat menjalankan dan menguji aplikasi *web* secara lokal tanpa memerlukan koneksi ke *server* produksi.

Penggunaan *XAMPP* sangat membantu dalam proses pengembangan dan pengujian sistem sebelum aplikasi dipublikasikan. Melalui *server* lokal, pengembang dapat melakukan simulasi penggunaan aplikasi, menguji fungsi-fungsi sistem, serta mendeteksi kesalahan sejak tahap awal pengembangan. Menurut Muludin dkk. (2025), *XAMPP* dapat digunakan sebagai lingkungan pengembangan lokal yang membantu pengembang dalam proses pembuatan dan pengujian aplikasi website sebelum diterapkan pada lingkungan sebenarnya.

Selain kemudahan pengujian, *XAMPP* juga menawarkan kepraktisan dalam hal konfigurasi *server*. Seluruh komponen utama sudah tersedia dalam satu paket sehingga pengembang tidak perlu melakukan instalasi dan pengaturan secara manual. Oleh karena itu, penggunaan *XAMPP* dalam pengembangan sistem informasi Sekolah Sepak Bola (SSB) sangat membantu dalam mempercepat proses

pengujian serta memastikan aplikasi berjalan dengan baik sebelum digunakan oleh pengguna sebenarnya.

2.2.7 MYSQL

MySQL merupakan sistem manajemen basis data relasional (*Relational Database Management System/RDBMS*) yang banyak digunakan dalam pengembangan aplikasi *web*. Sistem ini berfungsi untuk menyimpan, mengelola, dan memproses data dalam bentuk tabel-tabel yang saling berelasi. *MySQL* mendukung penggunaan *Structured Query Language (SQL)*, yaitu bahasa standar yang dirancang untuk melakukan pengelolaan dan manipulasi data pada basis data relasional (Sinlae dkk., 2024).

SQL memungkinkan pengembang untuk melakukan berbagai operasi basis data secara terstruktur dan efisien. Operasi tersebut meliputi pemilihan data (*select*), penyisipan data (*insert*), pembaruan data (*update*), serta penghapusan data (*delete*) yang dapat dilakukan secara mudah dan terotomatisasi. Dengan dukungan *SQL*, proses pengelolaan data dalam *MySQL* menjadi lebih sistematis dan mampu menangani kebutuhan data dalam jumlah besar.

Sebagai *server* basis data dengan konsep modern, *MySQL* menawarkan berbagai fitur unggulan yang mendukung kinerja dan keamanan sistem. *MySQL* memiliki keistimewaan dalam hal konektivitas, kecepatan pemrosesan, serta mekanisme keamanan yang baik untuk akses melalui jaringan *internet*. Menurut Siregar (2024), keandalan dan efisiensi yang dimiliki *MySQL* menjadikannya sangat cocok digunakan sebagai basis data pada aplikasi *web* berskala kecil hingga besar.

2.2.8 Composer

Composer merupakan alat manajemen dependensi yang digunakan dalam pengembangan aplikasi berbasis PHP untuk mengelola pustaka dan paket yang dibutuhkan oleh suatu proyek. Dalam konteks *framework* Laravel, *Composer* berperan sebagai pengelola pustaka utama yang memastikan seluruh dependensi aplikasi dapat terinstal dan terkelola dengan baik. Dengan menggunakan *Composer*, pengembang dapat menjaga konsistensi lingkungan pengembangan antar sistem

yang berbeda. *Composer* mempermudah proses instalasi dan pembaruan Laravel melalui perintah sederhana yang dijalankan melalui *command line*

Menurut *Laravel Official Documentation* (2023), “*Composer* adalah alat manajemen dependensi untuk PHP yang digunakan untuk mengelola semua pustaka dan *file* yang diperlukan oleh aplikasi Laravel.” Selain itu, *Composer* secara otomatis memastikan kompatibilitas antar-*paket* sehingga risiko konflik dependensi dapat diminimalkan.

Selain kemudahan instalasi, *Composer* juga menyediakan berkas *composer.json* yang berfungsi sebagai pusat pengelolaan dependensi proyek. Berkas ini mencatat seluruh pustaka yang digunakan sehingga pengelolaan dan pemeliharaan aplikasi menjadi lebih terorganisir dan efisien. Dengan *Composer*, pengembang dapat menambahkan berbagai fitur tambahan pada aplikasi Laravel, seperti autentikasi, pengelolaan basis data, maupun integrasi *API*, tanpa harus melakukan konfigurasi secara manual.

2.2.9 Performance Testing

Performance testing merupakan jenis pengujian perangkat lunak yang bertujuan untuk memastikan bahwa sistem dapat beroperasi secara optimal di bawah beban kerja tertentu. Pengujian ini tidak berfokus pada pencarian kesalahan (*bug*), melainkan pada pengukuran dan evaluasi kinerja sistem secara keseluruhan. Menurut Permatasari (2020), *performance testing* dilakukan untuk mengidentifikasi serta menghilangkan *performance bottleneck* yang dapat memengaruhi efektivitas sistem.

Fokus utama dalam *performance testing* mencakup tiga aspek penting, yaitu kecepatan (*speed*), skalabilitas (*scalability*), dan stabilitas (*stability*). Aspek kecepatan bertujuan untuk mengukur waktu respons aplikasi dalam menangani setiap permintaan pengguna. Dengan pengujian ini, pengembang dapat mengetahui apakah sistem mampu memberikan respons yang cepat sesuai dengan standar yang ditetapkan.

Selain itu, aspek skalabilitas dan stabilitas juga menjadi perhatian utama dalam *performance testing*. Skalabilitas mengukur kemampuan aplikasi dalam

menangani peningkatan jumlah pengguna atau beban kerja maksimum, sedangkan stabilitas menilai kestabilan sistem ketika dijalankan dalam berbagai kondisi beban. Oleh karena itu, *performance testing* berperan penting dalam memastikan aplikasi tetap andal, responsif, dan stabil pada berbagai kondisi operasional.

2.2.10 Unit Testing

Unit testing merupakan metode pengujian perangkat lunak yang bertujuan untuk memastikan bahwa setiap unit atau bagian terkecil dari kode program dapat berfungsi dengan benar. Pengujian ini dilakukan pada level fungsi atau modul secara terpisah untuk memastikan hasil yang diharapkan sesuai dengan spesifikasi. Dengan *unit testing*, kualitas kode dapat dijaga sejak tahap awal pengembangan perangkat lunak.

Dalam proses pengembangan perangkat lunak, sering ditemukan bagian kode yang tidak sesuai, tidak optimal, atau bahkan tidak terpakai. Melalui *unit testing*, pengembang dapat mengidentifikasi kode yang bermasalah serta mengevaluasi kinerja dan efisiensi kode tersebut. Hal ini membantu pengembang dalam melakukan perbaikan lebih awal sebelum kode digabungkan ke dalam sistem yang lebih besar.

Selain itu, *unit testing* juga berperan penting dalam mendeteksi kesalahan dan *bug* yang sulit ditemukan saat aplikasi dijalankan secara keseluruhan. Menurut Hasibuan dan Dirgahayu (2020), *unit testing* mampu mengungkap kesalahan tersembunyi yang tidak terlihat pada pengujian tingkat sistem. Oleh karena itu, penerapan *unit testing* dapat meningkatkan keandalan dan kualitas perangkat lunak secara menyeluruh.