

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait Penelitian

Tahapan telaah pustaka pada penelitian ini ditujukan untuk mengekstraksi wawasan dari berbagai riset terdahulu, yang difungsikan sebagai tolok ukur komparatif sekaligus fondasi empiris dalam proses rekayasa perangkat lunak. Kajian terhadap literatur-literatur yang relevan ini dinilai krusial untuk memastikan bahwa arah pengembangan sistem tetap selaras dengan urgensi faktual di lapangan serta terhindar dari repetisi studi yang telah ada. Lebih jauh, eksplorasi kepustakaan ini secara spesifik difokuskan pada tiga domain utama yang menjadi pilar kerangka pemikiran, yakni arsitektur sistem informasi sekolah terpadu, implementasi *Learning Management System* (LMS), dan digitalisasi manajemen keuangan siswa. Melalui sintesis komprehensif dari ketiga domain tersebut, penelitian ini diharapkan mampu memetakan batasan teknologi yang ada saat ini sekaligus mengidentifikasi secara tajam celah penelitian (*research gap*) yang dapat diisi oleh inovasi sistem yang diusulkan. Berikut adalah uraian mengenai studi-studi esensial yang dijadikan landasan rujukan:

2.1.1 Kajian Penelitian Terdahulu

Penelitian oleh Panyili dkk. (2024) [15] mengenai pengembangan Learning Management System (LMS) berbasis website membuktikan bahwa sistem ini efektif meningkatkan kualitas pembelajaran dan memudahkan akses materi siswa. Namun, sistem yang dikembangkan masih berdiri sendiri. Ketiadaan integrasi dengan sistem keuangan membuat pihak sekolah harus mengelola administrasi melalui platform yang terpisah.

Kajian literatur oleh [Nama Penulis] (Tahun) [16] pada Jurnal Paedagogie juga menegaskan bahwa LMS berkontribusi besar terhadap kemandirian dan hasil belajar siswa. Meskipun demikian, kajian tersebut murni berfokus pada aspek pedagogik. Pada praktiknya di SMK, pemisahan sistem akademik dan keuangan justru menimbulkan beban kerja ganda bagi guru dan staf, memicu fragmentasi data, serta inefisiensi operasional.

Lebih lanjut, Armilda Agripina Shafa (2024) [17] menunjukkan bahwa implementasi LMS mampu mengoptimalkan pengelolaan tugas dan komunikasi pembelajaran. Akan tetapi, penelitian ini memiliki keterbatasan yang sama, yakni belum menyentuh pengembangan sistem informasi terpadu yang menghubungkan layanan akademik dengan administrasi operasional sekolah.

Tabel 2. 1 Ringkasan Penelitian Terdahulu

No.	Penelitian dan Tahun	Judul singkat	Teknologi	Persamaan	Perbedaan dan Kekurangan
1.	Abidin dkk. (2023)	Sistem Keuangan Sekolah + WA Gateway[18]	PHP, MySQL, WA Gateway	Modul keuangan berbasis <i>web</i>	Tidak ada modul LMS; tidak menggunakan Laravel 12
2.	Susilawati (2022)	Pembayaran SPP Berbasis SMS Gateway[24]	PHP, SMS Gateway	Pencatatan SPP digital	Tidak ada LMS; tidak terintegrasi
3.	Panyili dkk. (2024)	LMS Berbasis Website SMK[15]	Web, PHP MySQL,	LMS berbasis <i>web</i> untuk SMK	Tidak ada modul keuangan terintegrasi
4.	Armilda Agripina Shafa. (2024).	Implementasi LMS di SMK[17]	Web, PHP MySQL,	LMS + pengujian SUS untuk SMK	Tidak ada integrasi modul keuangan

No.	Penelitian dan Tahun	Judul singkat	Teknologi	Persamaan	Perbedaan dan Kekurangan
5.	Nasution & Maulana (2024)	SIA Berbasis Laravel di SMK[25]	Laravel, MySQL	Laravel untuk lingkungan SMK	Hanya akademik; tanpa LMS & keuangan
6.	Wahyuni dkk. (2025)	SIM Sekolah Terpadu Berbasis Web[13]	Web, PHP MySQL,	Sistem informasi sekolah terpadu	Tidak mencakup LMS & keuangan siswa
7.	Akhkam (2026) Penelitian Ini	Sistem Informasi Sekolah Terpadu: Integrasi LMS + Keuangan Siswa (SMK Hasyim Asy'ari Bojong)	Laravel 12, Vue.js 3, Vuetify 3, Pinia, Vite, MySQL	Mengintegrasikan LMS + Keuangan dalam satu platform	Kontribusi baru: Integrasi penuh LMS + Keuangan + Black Box di SMK dengan teknologi terkini

Berdasarkan tabel perbandingan di atas, dapat disimpulkan bahwa belum ada penelitian terdahulu yang secara khusus dan komprehensif mengintegrasikan modul *Online Learning Management System* (LMS) dengan modul Sistem Manajemen Keuangan Siswa dalam satu *platform* berbasis *web* untuk lingkungan SMK menggunakan kombinasi teknologi modern Laravel 12 dan Vue.js 3. Penelitian ini mengisi kesenjangan tersebut dengan mengembangkan prototipe Sistem Informasi Sekolah Terpadu di SMK Hasyim Asy'ari Bojong yang diuji secara komprehensif menggunakan *Black Box Testing* memberikan kontribusi ilmiah yang terukur dan relevan dengan kebutuhan nyata di lapangan.

2.1.2 Analisis Kesenjangan Penelitian (Research Gap)

Berdasarkan kajian terhadap seluruh penelitian terdahulu yang telah diuraikan pada subbab-subbab sebelumnya, dapat diidentifikasi kesenjangan penelitian (*research gap*) yang signifikan. Penelitian - penelitian tentang sistem informasi manajemen keuangan sekolah berbasis *web*, seperti yang dilakukan oleh Abidin dkk. (2023), Rosalina dkk. (2025), dan Sidiq dkk. (2024), hanya berfokus pada penyelesaian permasalahan keuangan saja tanpa mengintegrasikannya dengan pengelolaan pembelajaran digital. Sebaliknya, penelitian-penelitian tentang LMS, seperti yang dilakukan oleh Panyili dkk.

(2024), Hariyanti dkk. (2024), dan Jaya dkk. (2024), hanya berfokus pada peningkatan efektivitas pembelajaran tanpa menyentuh aspek keuangan sekolah. Kondisi dikotomis ini mencerminkan bahwa belum ada penelitian yang secara khusus dan komprehensif mengintegrasikan kedua modul tersebut dalam satu platform terpadu.

Di sisi lain, penelitian-penelitian tentang sistem informasi sekolah terpadu, seperti yang dilakukan oleh Nasution dan Maulana (2024) serta Wahyuni dkk. (2025), meskipun menawarkan konsep keterpaduan sistem, tetap tidak mencakup integrasi antara LMS dan manajemen keuangan siswa secara bersamaan dalam satu platform. Dari aspek metode pengembangan, penelitian-penelitian yang ada menunjukkan bahwa metode *prototype*, Agile, dan Waterfall masing-masing memiliki kelebihan dan kekurangan dalam konteks pengembangan sistem informasi sekolah. Metode *prototype* dinilai paling sesuai untuk pengembangan prototipe sistem informasi yang kompleks dengan kebutuhan pengguna yang perlu diverifikasi secara iteratif.

Dengan demikian, kesenjangan penelitian yang diisi oleh penelitian ini adalah:

1. Belum adanya penelitian yang mengintegrasikan modul LMS dan sistem manajemen keuangan siswa dalam satu platform berbasis *web* khusus untuk SMK

2. Belum adanya penelitian yang menggunakan kombinasi teknologi Laravel 12 dan *Vue.js* 3 dalam pengembangan sistem informasi sekolah terpadu
3. Belum adanya penelitian sistem informasi sekolah terpadu yang diuji secara komprehensif menggunakan *Black Box Testing* dan UAT di lingkungan SMK di wilayah Kabupaten Tegal. Ketiga kesenjangan tersebut secara langsung dijawab oleh penelitian ini melalui pengembangan Sistem Informasi Sekolah berbasis *web* di SMK Hasyim Asy'ari Bojong.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Eksistensi sistem informasi menempati posisi sentral dalam tata kelola data maupun proses penentuan kebijakan strategis pada institusi masa kini. Secara konseptual, sistem ini direpresentasikan sebagai konstelasi elemen yang saling terintegrasi untuk menghimpun, mengolah, merawat, hingga menyalurkan arus data demi memfasilitasi langkah pengambilan keputusan, sinkronisasi manajerial, pengawasan, hingga representasi visual kinerja suatu entitas. Pemahaman ini menggarisbawahi bahwa sistem informasi tidak dapat direduksi hanya sebagai wujud peranti lunak (aplikasi), melainkan sebuah sinergi

holistik yang mengolaborasikan sumber daya manusia, infrastruktur teknologi, basis data, dan tata laksana operasional[26].

Ditinjau melalui kacamata implementasi, ekosistem informasi juga diklasifikasikan sebagai rantai komponen yang beroperasi secara kohesif dalam menjalankan fungsi ekstraksi, pemeliharaan, pemrosesan, dan distribusi data krusial untuk menopang ketepatan langkah taktis sebuah organisasi[27]. Relevansinya dengan rancang bangun prototipe ini adalah bahwa penguasaan komprehensif terhadap landasan teoritis sistem informasi berperan sebagai pilar utama dalam merumuskan arsitektur teknologi yang terpadu, presisi, dan sepenuhnya adaptif terhadap dinamika kebutuhan administratif di SMK Hasyim Asy'ari Bojong.

a. Komponen Sistem Informasi

Sistem informasi tersusun dari beberapa komponen utama yang saling mendukung[26]. Mengidentifikasi komponen sistem informasi terdiri dari:

1. Perangkat Keras (*Hardware*), yaitu perangkat fisik yang digunakan untuk memproses dan menyimpan data
2. Perangkat Lunak (*Software*), yaitu sekumpulan instruksi yang mengendalikan perangkat keras
3. Data, yaitu fakta-fakta mentah yang dikelola oleh sistem

4. Prosedur, yaitu aturan dan kebijakan yang mengatur penggunaan sistem
5. Sumber Daya Manusia (*Brainware*), yaitu individu yang mengoperasikan dan mengelola sistem
6. Jaringan Komunikasi, yaitu infrastruktur yang menghubungkan seluruh komponen sistem

b. Fungsi Sistem Informasi

Menurut beberapa literatur, sistem informasi dalam organisasi menjalankan tiga fungsi utama, yaitu mendukung operasi harian (*operational support*), menyediakan informasi bagi manajemen untuk pengambilan keputusan (*management support*), dan mendukung pencapaian keunggulan atau tujuan strategis organisasi. Dalam konteks penelitian ini, sistem informasi sekolah terpadu yang dikembangkan menjalankan ketiga fungsi tersebut: fungsi operasional melalui pengelolaan kegiatan pembelajaran dan transaksi keuangan harian, fungsi manajerial melalui penyediaan *dashboard monitoring* bagi kepala sekolah dan pengelola, serta fungsi strategis melalui dukungan terhadap peningkatan kualitas layanan pendidikan di SMK Hasyim Asy'ari Bojong secara berkelanjutan

2.2.2 Sistem Informasi Sekolah

Sistem informasi sekolah merupakan penerapan khusus dari konsep sistem informasi dalam konteks manajemen dan administrasi lembaga pendidikan. mendefinisikan sistem informasi manajemen sekolah berbasis *web* sebagai suatu platform digital yang mengintegrasikan berbagai modul administrasi pendidikan, mulai dari pengelolaan data siswa, jadwal pembelajaran, penilaian, hingga keuangan, dalam satu lingkungan sistem yang terpusat dan dapat diakses secara daring [13]. Sistem ini dirancang untuk menggantikan proses-proses manual yang rawan kesalahan dan tidak efisien dengan proses digital yang terstruktur, akurat, dan dapat diakses secara real-time oleh seluruh pemngku kepentingan sekolah.

2.2.3 Learning Management System (LMS)

Learning Management System (LMS) diklasifikasikan sebagai ruang wacana digital yang direkayasa secara khusus guna mengakomodasi, menata, dan menyebarluaskan substansi akademik melalui satu pusat komando yang terpadu. Infrastruktur ini memberikan keleluasaan bagi tenaga pendidik dalam mempublikasikan modul ajar, menyusun kerangka kurikulum, mengonfigurasi kalender akademik, hingga mendistribusikan instrumen penugasan secara lebih sistematis. Lebih jauh, kapabilitas LMS juga merambah pada ranah evaluasi

formatif maupun sumatif melalui penyediaan fitur kuis, ujian berbasis komputer, kalkulasi skor otomatis, hingga rekapitulasi kehadiran peserta didik secara seketika (*real-time*).

Sebagaimana ditegaskan dalam literatur terkait, LMS merepresentasikan sebuah infrastruktur digital yang dikonstruksi secara spesifik untuk mengorganisasi dan menyalurkan materi edukasi melalui satu pintu utama yang terstruktur[29]. Pemanfaatan *platform* ini membuka ruang bagi guru untuk merespons hasil kerja siswa melalui umpan balik (*feedback*) presisi, mengawasi kurva progres belajar, hingga memoderasi diskursus interaktif antarsiswa. Eksistensi modul-modul lanjutan, semacam wadah diskusi virtual, konferensi video, analitik performa belajar, hingga rekam jejak capaian akademik, turut mengamplifikasi kualitas pengalaman pembelajaran jarak jauh.

Secara keseluruhan, LMS kini telah bermetamorfosis menjadi suatu ekosistem pedagogi digital yang holistik. Peran *platform* ini melingkupi tiap fase dalam siklus Kegiatan Belajar Mengajar (KBM), yang membentang dari tahap desain perencanaan, eksekusi instruksional, parameter evaluasi, hingga mekanisme tindak lanjut pengayaan. Eksistensi teknologi ini secara nyata berkontribusi dalam memperluas jangkauan pendidikan, mendobrak batasan ruang dan waktu belajar, serta mereduksi kendala manajemen kelas sebuah

transformasi manajerial vital yang kini menjadi tulang punggung operasional pendidikan pascapandemi.

a. Manfaat LMS Dalam Pendidikan Vokasi

Implementasi Learning Management System (LMS) di lingkungan pendidikan, khususnya pada tingkat Sekolah Menengah Kejuruan (SMK), memberikan manfaat yang signifikan bagi seluruh pemangku kepentingan. Berbagai penelitian menunjukkan bahwa penggunaan LMS berbasis *web* mampu meningkatkan motivasi belajar, keterlibatan siswa, dan efektivitas proses pembelajaran, karena LMS menyediakan fitur pengelolaan materi, penugasan proyek, forum diskusi, serta penilaian secara daring yang terstruktur[30]. Manfaat utama penerapan *Learning Management System (LMS)* dalam pendidikan meliputi:

1. Fleksibilitas akses materi pembelajaran kapan saja dan di mana saja bagi siswa.
2. Kemudahan pemantauan kemajuan belajar melalui laporan dan nilai yang tersaji secara *real-time*.
3. Efisiensi distribusi materi dan penugasan karena seluruh aktivitas pembelajaran terdokumentasi dalam satu *platform*.
4. Peningkatan interaktivitas antara guru dan siswa melalui forum diskusi, kuis, dan tugas daring.

5. Dokumentasi aktivitas pembelajaran yang lebih rapi dan mudah dievaluasi oleh pihak sekolah.

Selain itu, untuk menilai tingkat kenyamanan penggunaan LMS, berbagai penelitian menggunakan *System Usability Scale* (SUS) sebagai instrumen standar dalam mengukur aspek kemudahan penggunaan, efisiensi, dan kepuasan pengguna terhadap sistem pembelajaran digital[31].

b. Fitur dan Komponen LMS

Fitur-fitur utama yang umumnya terdapat dalam sebuah LMS mencerminkan kebutuhan pengelolaan pembelajaran yang komprehensif mengidentifikasi fitur inti LMS meliputi: manajemen kelas virtual, distribusi dan unduhan materi ajar, sistem penugasan (*assignment*) daring, kuis dan ujian *online*, presensi digital, penilaian berbasis rubrik, serta forum diskusi interaktif antara guru dan siswa[15]. Dalam penelitian ini, modul LMS yang dikembangkan mencakup seluruh fitur tersebut, yaitu:

1. Manajemen kelas virtual dengan pengelompokan mata pelajaran dan kelas
2. Unggah dan unduh materi pembelajaran dalam berbagai format
3. Sistem penugasan dengan batas waktu pengumpulan
4. Kuis *online* dengan penilaian otomatis

5. Presensi digital berbasis sistem
6. Serta rekap nilai dan laporan akademik terintegrasi.

2.2.4 Sistem Informasi Manajemen Keuangan

Sistem informasi manajemen keuangan sekolah adalah subsistem yang bertanggung jawab atas pengelolaan seluruh transaksi keuangan di lingkungan sekolah secara digital dan terstruktur mendefinisikan sistem manajemen keuangan sekolah berbasis *web* sebagai sistem yang mampu mengelola, mencatat, memverifikasi, dan melaporkan seluruh transaksi keuangan sekolah secara otomatis dengan antarmuka yang dapat diakses oleh staf administrasi, kepala sekolah, dan orang tua siswa[18]. Digitalisasi manajemen keuangan sekolah merupakan langkah krusial dalam meningkatkan transparansi, akurasi, dan akuntabilitas pengelolaan keuangan lembaga pendidikan.

a. Komponen Sistem Keuangan Sekolah

Sistem informasi manajemen keuangan sekolah pada umumnya mencakup beberapa komponen utama yang saling terintegrasi. Mengidentifikasi komponen-komponen tersebut meliputi:

1. Modul Penerimaan: yang mencakup pencatatan seluruh pemasukan sekolah dari berbagai sumber

2. Modul Pembayaran Siswa: yang mengelola pencatatan SPP dan iuran lainnya per siswa
3. Modul Pelaporan: yang menghasilkan laporan keuangan per periode secara otomatis
4. Modul Monitoring: yang menyediakan *dashboard* informasi keuangan secara *real-time* bagi kepala sekolah

b. Pencatatan SPP dan Pelaporan Keuangan

Pencatatan Sumbangan Pembinaan Pendidikan (SPP) merupakan Pencatatan Sumbangan Pembinaan Pendidikan (SPP) merupakan salah satu proses penting dalam administrasi keuangan sekolah. Namun, pencatatan yang masih dilakukan secara manual cenderung memakan waktu, rentan terhadap kesalahan, dan kurang efisien karena seluruh proses mulai dari pembayaran hingga pelaporan dilakukan menggunakan buku pencatatan. Selain meningkatkan beban kerja administrasi, proses manual juga menyebabkan kurangnya transparansi pengelolaan keuangan kepada orang tua atau wali siswa.[32].

Salah satu permasalahan utama dalam sistem manual adalah tingginya risiko *human error*, seperti kesalahan input pembayaran atau keterlambatan pembaruan status pembayaran siswa. Selain itu, data pembayaran dan laporan keuangan yang tersimpan di berbagai

dokumen atau *file* terpisah menyebabkan proses pencarian, verifikasi, dan pembuatan laporan menjadi lebih lambat dan tidak efisien.[33].

Permasalahan serupa juga terjadi di berbagai sekolah, seperti penggunaan buku besar pembayaran yang rentan rusak, kehilangan kartu pembayaran siswa, hingga risiko kehilangan data keuangan. Kondisi tersebut relevan dengan permasalahan yang ditemukan di SMK Hasyim Asy'ari Bojong. Oleh karena itu, penelitian ini mengembangkan modul manajemen keuangan berbasis *web* yang mampu mengotomatisasi proses pencatatan SPP, verifikasi pembayaran, pelaporan, dan *monitoring* pembayaran siswa secara *real-time* guna meningkatkan efisiensi, akurasi, dan transparansi pengelolaan keuangan sekolah.

2.2.5 Sistem Informasi Terpadu

Sistem informasi terpadu (*integrated information system*) adalah pendekatan pengembangan sistem yang menggabungkan beberapa modul atau subsistem yang sebelumnya berjalan secara terpisah ke dalam satu platform yang saling terhubung, berbagi data secara konsisten, dan dikelola melalui satu antarmuka yang terintegrasi. Sistem administrasi yang masih bersifat manual, terfragmentasi, dan

tidak terintegrasi terbukti menjadi penghambat utama dalam efisiensi operasional dan pengambilan keputusan, karena sistem seperti ini sering kali menimbulkan redundansi data, keterlambatan informasi, dan meningkatnya risiko kesalahan manusia. Organisasi modern membutuhkan sistem administrasi digital yang terintegrasi, mampu mengotomatiskan tugas rutin, menyajikan data secara *real-time*, serta mendorong kolaborasi lintas unit kerja[35].

Kondisi fragmentasi sistem teknologi informasi di sekolah menengah, di mana *Learning Management System (LMS)* dan sistem administrasi akademik seperti pengolahan nilai, kehadiran, dan keuangan berjalan secara terpisah dan tidak terintegrasi, menimbulkan inefisiensi operasional, duplikasi data, dan beban kerja ganda bagi guru dan tenaga administrasi, sehingga menghambat efektivitas proses pendidikan. Kondisi inilah yang secara langsung menjadi permasalahan utama yang diangkat dalam penelitian ini di SMK Hasyim Asy'ari Bojong, sehingga diperlukannya sistem informasi terpadu yang mampu menyatukan seluruh proses administrasi sekolah dalam satu platform yang saling terhubung.

a. Konsep Integrasi Sistem

Integrasi sistem dalam konteks pengembangan perangkat lunak mengacu pada proses menghubungkan dua atau lebih sistem atau modul yang sebelumnya independen sehingga dapat

beroperasi sebagai satu kesatuan yang kohesif. Dalam penelitian ini, integrasi antara modul *Learning Management System* (LMS) dan modul manajemen keuangan siswa diwujudkan melalui arsitektur *RESTful API* yang memungkinkan kedua modul berbagi data secara real-time melalui satu basis data MySQL yang terpusat[36].

Implementasi sistem berbasis *web* dengan arsitektur *RESTful API* menggunakan *framework* Laravel telah terbukti mampu menghasilkan sistem yang responsif dan efisien, dengan tingkat interoperabilitas yang tinggi antar-modul administrasi akademik. Hal ini menunjukkan bahwa pemanfaatan *RESTful API* sebagai jembatan komunikasi antar-modul dapat meningkatkan keterpaduan, konsistensi data, serta kinerja sistem yang berbasis *web*, baik untuk lingkungan pendidikan maupun administrasi umum.

b. Manfaat Integrasi LMS dan Manajemen Keuangan

Integrasi antara modul LMS dan sistem manajemen keuangan siswa dalam satu platform memberikan manfaat yang sangat signifikan dibandingkan sistem yang berjalan terpisah menyatakan bahwa sistem informasi manajemen sekolah yang terintegrasi mampu menghilangkan redundansi data, meningkatkan akurasi informasi, mempermudah pemantauan lintas modul secara real-

time, serta mendukung pengambilan keputusan manajerial yang lebih cepat dan berbasis data. Secara spesifik, integrasi LMS dengan keuangan dalam penelitian ini memungkinkan:

1. Pengelolaan seluruh kebutuhan akademik dan keuangan dalam satu platform
2. Eliminasi fragmentasi data antara sistem pembelajaran dan keuangan
3. Akses informasi yang komprehensif bagi seluruh pemangku kepentingan
4. Efisiensi operasional yang lebih tinggi bagi staf administrasi dan tenaga pendidik

2.2.6 Aplikasi Berbasis *Web*

Aplikasi berbasis *web* (*web-based application*) adalah perangkat lunak yang dijalankan melalui *browser* internet dan dapat diakses melalui jaringan komputer tanpa perlu menginstal aplikasi tambahan pada perangkat pengguna. Aplikasi ini beroperasi dengan memanfaatkan arsitektur *client-server*, di mana sisi *client* (*browser*) bertugas menampilkan antarmuka kepada pengguna, sedangkan pemrosesan data dan logika bisnis dilakukan di sisi *server*.

Aplikasi berbasis *web* merupakan sistem yang berjalan di atas infrastruktur *web* dengan memanfaatkan arsitektur *client-server*, di

mana pengguna mengakses layanan melalui browser sementara pemrosesan data dilakukan di sisi *server*. Keunggulan utama aplikasi berbasis *web* mencakup aksesibilitas yang tinggi dari berbagai perangkat (*cross-platform*), kemudahan pemeliharaan dan pembaruan sistem yang cukup dilakukan di satu tempat (*server*), serta efisiensi biaya distribusi yang jauh lebih rendah dibandingkan dengan aplikasi *desktop* atau *mobile native*.

Selain itu, aplikasi berbasis *web* juga menawarkan kemudahan kolaborasi, skalabilitas yang baik, serta kemampuan untuk selalu memberikan versi terbaru kepada seluruh pengguna secara bersamaan tanpa proses update manual di setiap perangkat[37].

a. Arsitektur *Client-Server*

Arsitektur *client-server* merupakan model komunikasi dalam jaringan komputer di mana *client* mengirimkan permintaan kepada *server* yang kemudian memproses permintaan tersebut dan mengembalikan respons. Dalam sistem yang dikembangkan penelitian ini, arsitektur *client-server* diimplementasikan dengan *Vue.js 3* sebagai client yang berkomunikasi dengan *backend Laravel 12* melalui *RESTful API*, dengan *MySQL* sebagai basis data terpusat membuktikan bahwa implementasi *Laravel* sebagai *REST server* dan *Vue.js* sebagai *REST client* memiliki interoperabilitas yang baik dengan performa tinggi[38].

b. Responsive Web Design

Responsive Web Design (RWD) adalah pendekatan perancangan antarmuka *web* yang memastikan tampilan sistem menyesuaikan diri secara otomatis dengan ukuran layar perangkat yang digunakan, baik *desktop*, tablet, maupun *smartphone*. Menyimpulkan bahwa platform *web* merupakan pilihan teknologi yang paling tepat untuk sistem informasi sekolah karena memungkinkan seluruh pengguna untuk mengakses sistem dari perangkat apapun tanpa batasan sistem operasi[25]. Dalam penelitian ini, pendekatan RWD diimplementasikan melalui penggunaan *Vuetify 3* yang menyediakan komponen *UI* responsif berbasis *Material Design*, memastikan sistem dapat diakses dengan nyaman baik melalui perangkat *desktop* di ruang guru maupun melalui *smartphone* oleh siswa.

2.2.7 Basis Data

Basis data atau *database* adalah kumpulan data yang terorganisir secara sistematis dalam suatu media penyimpanan digital yang dapat diakses, dimanipulasi, dan dikelola secara efisien menggunakan perangkat lunak manajemen basis data[39]. Basis data merupakan komponen kritis dalam pengembangan sistem informasi berbasis *web* karena berfungsi sebagai pusat penyimpanan seluruh data sistem yang

harus dijaga integritasnya, konsistensinya, dan keamanannya. Dalam sistem informasi sekolah terpadu yang dikembangkan penelitian ini, basis data mengelola data yang sangat beragam, mulai dari data pengguna, data akademik (materi, tugas, nilai, presensi), hingga data keuangan (pembayaran SPP, riwayat transaksi, laporan keuangan).

a. Database Management System (DBMS)

Database Management System (DBMS) merupakan perangkat lunak yang digunakan untuk membuat, mengelola, menyimpan, serta mengakses basis data secara terstruktur dan sistematis. Pada DBMS relasional, data disusun ke dalam tabel-tabel yang saling berhubungan melalui penggunaan *primary key* dan *foreign key* sehingga hubungan antar data dapat dikelola dengan baik. Selain itu, proses normalisasi basis data hingga tingkat *Third Normal Form* (3NF) diterapkan untuk meminimalkan redundansi data serta menjaga integritas dan konsistensi data dalam sistem. Perancangan basis data yang baik, yang divisualisasikan melalui *Entity Relationship Diagram* (ERD), menjadi fondasi penting dalam pembangunan sistem informasi yang terorganisir, efisien, dan andal[40]

b. MySQL

MySQL adalah sistem manajemen basis data relasional (*Relational Database Management System / RDBMS*) yang bersifat *open source* dan paling banyak digunakan dalam pengembangan

aplikasi *web* di seluruh dunia. Dalam era digital saat ini, pengembangan *Database Management System* (DBMS) menggunakan *MySQL* memiliki peran penting dalam menyimpan, mengelola, dan mengakses data secara efisien pada berbagai sistem informasi berbasis *web*[41]. Efektivitas integrasi data akademik sangat bergantung pada ketepatan perancangan *Entity Relationship Diagram* (ERD), proses normalisasi hingga tingkat 3NF untuk menjaga integritas data, serta penggunaan *framework* modern seperti Laravel yang meningkatkan keamanan dan skalabilitas sistem. Pengujian berdasarkan standar ISO 25010 juga membuktikan bahwa sistem berbasis *MySQL* memiliki reliabilitas tinggi dengan waktu respons yang stabil[42]. Dalam penelitian ini, *MySQL* digunakan sebagai *DBMS* utama yang diakses oleh *backend* Laravel 12 melalui *Eloquent ORM*, memungkinkan pengelolaan data akademik dan keuangan yang kompleks secara efisien dan aman.

2.2.8 Pemodelan Sistem

Unified Modeling Language (UML) adalah bahasa pemodelan visual standar yang digunakan secara luas dalam rekayasa perangkat lunak untuk mendokumentasikan, menspesifikasikan, dan memvisualisasikan rancangan sistem informasi secara sistematis dan terstruktur. Penerapan *UML* membantu memperjelas hubungan antar

komponen sistem dan interaksi pengguna, menghasilkan desain yang lebih terstruktur dan mudah dipahami oleh seluruh pemangku kepentingan pengembangan[43]. Dalam penelitian ini, *UML* digunakan sebagai alat utama pemodelan sistem pada tahap *Quick Design* dalam metode *Prototype* terdapat empat jenis diagram *UML* yang digunakan dalam penelitian ini:

1. ***Use Case Diagram***: Diagram yang menggambarkan interaksi antara pengguna (aktor) dengan sistem. Dalam penelitian ini, *Use Case Diagram* menggambarkan interaksi 4 aktor utama (Administrator, Guru, Siswa, dan Staf TU) dengan seluruh fitur modul LMS dan modul manajemen keuangan siswa.
2. ***Activity Diagram***: Diagram yang menggambarkan alur proses atau aktivitas dalam sistem secara terurut. *Activity Diagram* digunakan untuk menggambarkan alur proses pengelolaan materi, pengumpulan tugas, pencatatan pembayaran SPP, dan proses-proses bisnis lainnya dalam sistem.
3. ***Sequence Diagram***: Diagram yang menggambarkan interaksi antar objek dalam sistem berdasarkan urutan waktu. *Sequence Diagram* digunakan untuk menggambarkan alur komunikasi antara *Frontend (Vue.js 3)* dan *Backend (Laravel 12)* melalui *RESTful API*.

4. Class Diagram: Diagram yang menggambarkan struktur kelas, atribut, dan metode dalam sistem berorientasi objek. *Class Diagram* digunakan untuk menggambarkan struktur model (*Eloquent Model*) dalam arsitektur *Laravel 12*.

a. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah alat pemodelan basis data yang digunakan untuk menggambarkan struktur data suatu sistem secara visual, mencakup entitas (tabel), atribut (kolom), dan relasi (hubungan) antar entitas. *ERD* menjadi fondasi perancangan skema basis data yang akan diimplementasikan dalam *MySQL*. Perancangan *ERD* yang baik menerapkan teknik normalisasi secara sistematis mulai dari *Unnormalized Form* (UNF) hingga *Third Normal Form* (3NF), yang terbukti berhasil menghilangkan anomali data pada proses penyisipan, pembaruan, dan penghapusan, sehingga menjamin integritas dan konsistensi data secara menyeluruh. Dalam penelitian ini, *ERD* dirancang untuk menggambarkan relasi antar entitas utama sistem, termasuk: User, Kelas, Mata Pelajaran, Materi, Tugas, Penilaian, Presensi, Pembayaran SPP, dan Riwayat Transaksi[44].

2.2.9 Metode Prototype

Pendekatan *Prototype* direpresentasikan sebagai salah satu paradigma rekayasa perangkat lunak di dalam ekosistem *Software Development Life Cycle (SDLC)*. Fokus utama metode ini terletak pada konstruksi purwarupa awal yang selanjutnya divalidasi dan dievaluasi secara berulang (iteratif) bersama target pengguna hingga spesifikasi fungsionalnya berkesesuaian. Pendekatan ini dinilai sangat taktis untuk dieksekusi pada proyek sistem informasi yang elaborasi kebutuhan penggunaannya masih fluktuatif atau belum terpetakan secara ajek pada fase inisiasi. Hal ini dikarenakan kerangka kerja *Prototype* secara proaktif memfasilitasi dialog interaktif yang berkelanjutan antara pihak pengembang dan pengadopsi sistem sepanjang siklus pembuatan berlangsung.

Siklus pengembangan dalam metode ini beroperasi secara sirkuler hingga purwarupa dinyatakan sah dan layak oleh pengguna akhir. Rangkaian fase tersebut diawali dengan Ekstraksi Spesifikasi Kebutuhan (*Requirements Gathering*), yakni proses identifikasi mendalam terhadap parameter fungsional maupun non-fungsional sistem. Fase kedua berlanjut pada Perancangan Kerangka Awal (*Quick Design*) untuk memvisualisasikan arsitektur dasar aplikasi. Selanjutnya, masuk pada fase Konstruksi Purwarupa (*Build Prototype*) sebagai wujud implementasi pengkodean teknis dari rancangan

tersebut. Purwarupa yang telah berdiri kemudian memasuki fase Pengujian dan Penilaian (*Prototype Evaluation*) yang melibatkan interaksi pengguna secara langsung. Siklus ini ditutup dengan Penyempurnaan Purwarupa (*Refining Prototype*), yaitu saat sistem direvisi secara berkala berdasarkan umpan balik (*feedback*) pengguna sebelum beranjak pada putaran iterasi berikutnya[20].

Komparasi metode *Prototype* dengan model konvensional lainnya menyingkap sejumlah keunggulan komparatif yang menjadikannya sangat relevan untuk arsitektur sistem informasi di institusi pendidikan. Keunggulan pertama berakar pada pelibatan partisipatif pengguna sejak titik nol, yang menjadi garansi bahwa hasil akhir sistem akan linier dengan urgensi riil di lapangan. Keunggulan kedua bertumpu pada sifat iteratifnya yang memungkinkan deteksi dini terhadap anomali sistem, sehingga mereduksi probabilitas kegagalan proyek secara signifikan. Di samping itu, kehadiran purwarupa fisik yang dapat langsung dioperasikan memberikan pemahaman yang jauh lebih komprehensif bagi pemangku kepentingan dibandingkan dengan sekadar menelaah dokumen spesifikasi tertulis[21]. Konstelasi kelebihan tersebut mengukuhkan metode *Prototype* sebagai opsi paling rasional untuk memandu penelitian ini, mengingat tingginya kompleksitas integrasi antara dua pangkalan operasional raksasa yakni modul *LMS* dan tata kelola

keuangan yang melibatkan variasi kebutuhan dari berbagai spektrum pengguna.

2.2.10 Pengujian Sistem

Black Box Testing merupakan pendekatan evaluasi perangkat lunak yang menitikberatkan pada validasi fungsionalitas operasional dari sudut pandang pengguna akhir (*end-user*), tanpa perlu membedah arsitektur internal maupun logika *source code* (sintaksis program). Prosedur yang kerap diistilahkan sebagai pengujian perilaku (*behavioral testing*) ini berpijak sepenuhnya pada matriks spesifikasi kebutuhan sistem. Dalam eksekusinya, analis tidak dituntut untuk memahami alur komputasi di balik layar, melainkan cukup memberikan ragam simulasi masukan (*input*) untuk mengamati dan mengonfirmasi apakah respons (*output*) yang disajikan oleh sistem telah selaras dengan parameter fungsional yang dirumuskan sejak fase perancangan awal [45].

Pada ranah implementasi riset ini, instrumen *Black Box Testing* dioperasikan untuk mengesahkan integritas operasional pada seluruh fitur vital aplikasi. Cakupan evaluasi membentang dari mekanisme autentikasi akses dan otorisasi pengguna, fungsionalitas menyeluruh pada modul *Learning Management System* (yang mencakup sirkulasi

materi ajar, distribusi tugas, pelaksanaan kuis daring, pencatatan presensi, hingga kalkulasi nilai), sampai pada validasi modul tata kelola kas siswa (pencatatan setoran SPP, ekstraksi laporan keuangan, dan histori transaksi) beserta instrumen pemantauan pada *dashboard*. Seluruh rangkaian dokumentasi pengujian ini selanjutnya direkapitulasi melalui format matriks *test case*, yang merinci skenario interaksi, variabel data masukan, ekspektasi keluaran (*expected result*), hingga realitas respons sistem (*actual result*) yang terjadi di lapangan.

2.2.11 RESTful API

RESTful API (*Representational State Transfer Application Programming Interface*) merupakan gaya arsitektur komunikasi antar sistem berbasis *web* yang menggunakan protokol HTTP sebagai media pertukaran data, umumnya dalam format *JSON* maupun *XML*. Konsep *REST* pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000 melalui disertasi doktoralnya. Layanan *web* yang menerapkan prinsip arsitektur *REST* dikenal sebagai *RESTful Web Service*.

RESTful API dirancang untuk memungkinkan komunikasi antar aplikasi secara fleksibel, ringan, dan mudah diintegrasikan. Prinsip utama *REST* meliputi *stateless*, yaitu setiap permintaan diproses secara

mandiri tanpa menyimpan status sebelumnya; *client-server separation*, yaitu pemisahan antara antarmuka pengguna dan server; serta *uniform interface*, yaitu penggunaan standar komunikasi yang konsisten. Selain itu, *RESTful API* memanfaatkan metode HTTP seperti *GET*, *POST*, *PUT*, *PATCH*, dan *DELETE* dalam proses pengelolaan data. Implementasi *RESTful API* terbukti mampu mendukung integrasi sistem berbasis *web* secara lebih efisien dan terstruktur, khususnya pada pengembangan aplikasi menggunakan *framework* Laravel[46].

Dalam konteks pengembangan sistem informasi terpadu berbasis *web*, *RESTful API* berperan sebagai jembatan komunikasi yang menghubungkan komponen *frontend* dan *backend* sistem. Hadinata dan Stianingsih (2024) dalam penelitian analisis performa *RESTful API* antara Laravel dan *framework* lain membuktikan bahwa Laravel merupakan pilihan *framework* yang sangat tepat sebagai backend server *RESTful API* karena kemudahan implementasi dan stabilitasnya[38]. Lebih lanjut, implementasi *RESTful API* menggunakan *Laravel* dan *Vue.js* terbukti memiliki kinerja baik dengan interoperabilitas yang tinggi. Seluruh komunikasi data antara *Vue.js 3 (frontend)* dan *Laravel 12 (backend)* dilakukan melalui *RESTful API* dengan format *JSON*, menggunakan *Laravel Sanctum* sebagai mekanisme autentikasi berbasis token yang aman.

2.2.12 Teknologi yang Digunakan

a. Teknologi *Backend*

1) Laravel 12

Laravel adalah *framework* pengembangan aplikasi *web* berbasis bahasa pemrograman PHP yang mengadopsi pola arsitektur *Model-View-Controller* (MVC) dan menyediakan berbagai fitur bawaan yang mempercepat proses pengembangan menjelaskan bahwa *Laravel* menyediakan fitur canggih seperti *Eloquent ORM* untuk abstraksi basis data, *Artisan CLI* untuk otomasi tugas, sistem *routing* yang ekspresif, *middleware* untuk pengelolaan keamanan dan autentikasi, serta sistem *migration* untuk pengelolaan skema basis data secara terstruktur[47]. sebagai versi yang digunakan dalam penelitian ini, menghadirkan peningkatan performa yang signifikan, dukungan *PHP 8.2* yang lebih modern, fitur keamanan yang lebih kuat melalui *Laravel Sanctum*, serta ekosistem paket yang sangat kaya melalui *Composer Package Manager*.

2) PHP (Hypertext Preprocessor)

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side* yang paling banyak digunakan dalam pengembangan aplikasi *web*. *PHP 8.2*, sebagai versi yang digunakan dalam

penelitian ini, menghadirkan berbagai peningkatan performa dan fitur dibandingkan versi sebelumnya, termasuk: *readonly classes*, *intersection types*, peningkatan performa JIT (*Just-In-Time Compilation*), serta perbaikan keamanan yang signifikan. Penggunaan PHP 8.2 bersama Laravel 12 menghasilkan kombinasi teknologi *backend* yang modern, aman, dan berperforma[47].

3) MySQL

MySQL telah dibahas secara komprehensif pada subbab 2.2.7. Dalam konteks teknologi yang digunakan, MySQL dipilih sebagai *DBMS* penelitian ini karena integrasi yang sangat baik dengan Laravel 12 melalui *Eloquent ORM*, kemudahan pengelolaan skema basis data melalui fitur *migration* Laravel, serta ketersediaan yang terintegrasi dalam lingkungan pengembangan Laragon yang digunakan dalam penelitian ini.

b. Teknologi Frontend

1) Vue.js

Vue.js merupakan *framework JavaScript* progresif yang digunakan untuk membangun antarmuka pengguna (*user interface*) berbasis *web* yang interaktif dan reaktif pada sisi *frontend*. *Vue.js* dirancang dengan pendekatan berbasis komponen (*component-based architecture*) sehingga memudahkan

pengembangan aplikasi *web* modern yang dinamis dan mudah dipelihara. Pada penelitian ini digunakan Vue.js versi 3 yang menghadirkan berbagai peningkatan dibandingkan versi sebelumnya, seperti *Composition API*, peningkatan performa melalui optimasi *Virtual DOM*, serta dukungan *TypeScript* yang lebih baik.

Implementasi Vue.js pada arsitektur aplikasi berbasis *REST API* terbukti mampu mendukung komunikasi data secara dinamis antara *frontend* dan *backend* sehingga menghasilkan tampilan antarmuka yang responsif dan fleksibel. Penggunaan Vue.js juga mempermudah proses integrasi dengan *backend Laravel* melalui *RESTful API* dalam pengembangan sistem informasi berbasis *web*[48].

2) Vuetify 3

Vuetify 3 adalah library komponen *UI (User Interface)* berbasis *Material Design* yang dirancang khusus untuk Vue.js 3. Vuetify menyediakan koleksi komponen siap pakai yang kaya, mulai dari tombol, formulir, tabel data, dialog, navigasi, hingga grafik, yang semuanya telah menerapkan prinsip-prinsip *Material Design* dari Google. Penggunaan Vuetify 3 dalam penelitian ini memungkinkan pengembangan antarmuka

pengguna yang konsisten secara visual, responsif di berbagai ukuran layar, dan profesional tanpa perlu membangun komponen *UI* dari awal, sehingga mempercepat proses pengembangan secara signifikan.

3) Pinia

Pinia adalah library state management resmi yang direkomendasikan untuk Vue.js 3 sebagai pengganti Vuex. *State management* adalah mekanisme untuk mengelola data aplikasi yang perlu diakses dan dibagikan oleh banyak komponen secara global. *Pinia* menawarkan *API* yang lebih sederhana dan intuitif dibandingkan *Vuex*, dukungan penuh *TypeScript*, ukuran *bundle* yang lebih kecil, serta kemampuan *devtools* yang lebih baik untuk debugging. Dalam sistem yang dikembangkan penelitian ini, *Pinia* digunakan untuk mengelola *state global* seperti data autentikasi pengguna, data kelas aktif, dan pengaturan sistem yang dibutuhkan secara konsisten di seluruh komponen aplikasi.

4) Vite

Vite adalah *build tool* generasi baru untuk pengembangan aplikasi *web* modern yang menawarkan kecepatan jauh lebih tinggi dibandingkan *bundler* tradisional seperti *Webpack*. Keunggulan utama Vite meliputi: *Hot Module Replacement*

(*HMR*) yang sangat cepat sehingga perubahan kode langsung terlihat di *browser* tanpa *reload* penuh, proses *build* produksi yang efisien menggunakan *Rollup*, serta konfigurasi yang minimal dan mudah dipahami. Dalam penelitian ini, Vite digunakan sebagai *build tool* untuk komponen *frontend* Vue.js 3, memastikan proses pengembangan yang cepat dan *build* produksi yang dioptimalkan.

c. Teknologi Pendukung

1) Laragon

Laragon adalah lingkungan pengembangan lokal (*local development environment*) yang komprehensif dan ringan untuk sistem operasi Windows, yang dirancang khusus untuk mempermudah pengembangan aplikasi *web* berbasis PHP. *Laragon* menyediakan seluruh infrastruktur yang dibutuhkan dalam satu paket terintegrasi, meliputi: *web server Apache/Nginx*, *MySQL* sebagai *DBMS*, *PHP* dengan berbagai versi yang dapat dipilih, serta *Node.js*. Keunggulan *Laragon* dibandingkan alternatif lain seperti *XAMPP* adalah kemudahan konfigurasi domain lokal otomatis, performa yang lebih ringan, serta dukungan yang sangat baik untuk pengembangan aplikasi berbasis *Laravel*. Dalam penelitian ini, *Laragon* digunakan

sebagai server lokal selama proses pengembangan dan pengujian sistem.

2) Visual Studio Code

Visual Studio Code (VS Code) adalah *source code editor* yang dikembangkan oleh Microsoft, yang saat ini menjadi editor paling populer di kalangan pengembang *web* di seluruh dunia. *VS Code* menyediakan berbagai fitur yang mendukung produktivitas pengembangan, meliputi: *IntelliSense* untuk autocomplete kode yang cerdas, *syntax highlighting* untuk berbagai bahasa pemrograman, integrasi *Git* untuk pengelolaan versi kode, terminal terintegrasi, serta ekosistem *extensions* yang sangat kaya. Dalam penelitian ini, VS Code digunakan sebagai editor utama dengan dukungan *extensions* seperti *Laravel Extension Pack*, *Volar* (untuk *Vue.js 3*), *PHP Intelephense*, dan *GitLens* yang mempercepat dan memudahkan proses pengembangan sistem.