

BAB II

KAJIAN LITERATUR

2.1 Sistem Informasi

Sistem informasi merupakan salah satu teknologi yang dibutuhkan untuk dapat memudahkan dan menemukan informasi yang dibutuhkan dan mengelola data dengan lebih efektif dan efisien [11]. Dalam konteks pendidikan, sistem informasi berperan penting dalam proses administrasi dan pengelolaan data yang kompleks, mencakup aktivitas akademik, keuangan, dan komunikasi antar pihak yang terlibat di dalam sekolah. Terlebih, setelah adanya sistem informasi akademik berbasis *web* yang sangat membantu dalam memberikan akses informasi yang jauh lebih mudah, akurat dan juga cepat [12].

Berdasarkan teori yang telah dijelaskan, sistem informasi memiliki peranan penting dalam mendukung pengelolaan data sekolah secara terintegrasi dan terstruktur. Pada penelitian ini, sistem informasi digunakan sebagai dasar pengembangan platform sekolah yang mampu membantu proses administrasi, pengelolaan data akademik, serta penyampaian informasi secara lebih cepat dan efisien. Dengan adanya sistem informasi, sekolah dapat meningkatkan kualitas layanan dan meminimalkan kesalahan dalam pengolahan data.

2.1.1 Sistem Informasi Akademik

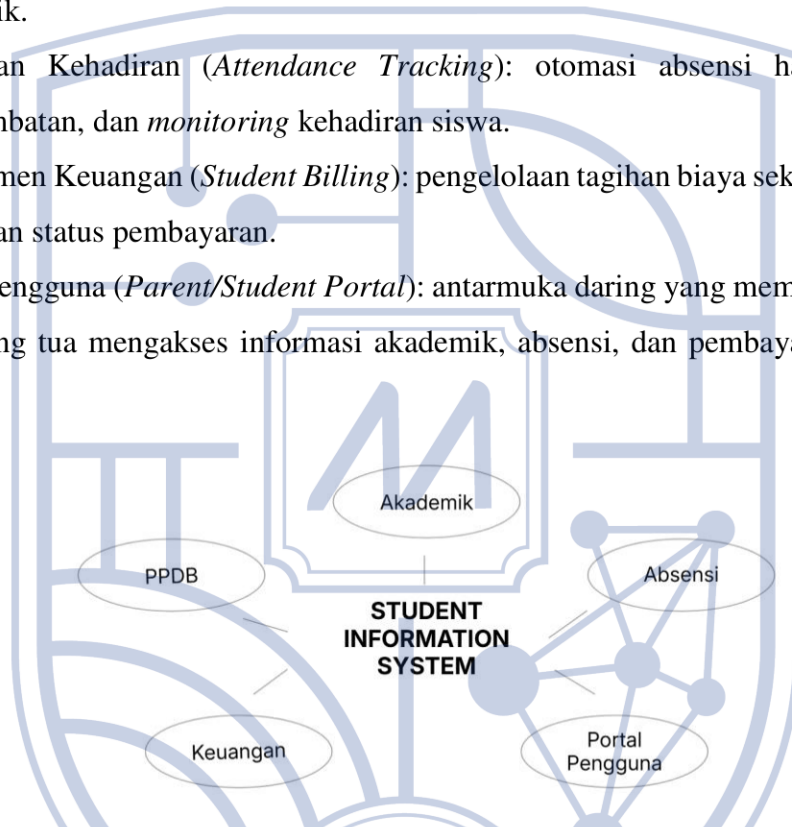
Sistem informasi akademik sering disebut dengan istilah *Student Information System* (SIS). SIS adalah platform berbasis *web* yang membantu sekolah atau perguruan tinggi mengelola data siswa secara *online* agar lebih mudah dikelola dan lebih jelas. SIS mengumpulkan berbagai data sekolah secara *online* sehingga dapat diakses dengan mudah oleh guru, orang tua, siswa, dan pihak administrasi. Data tersebut mencakup informasi pribadi siswa, nilai, hasil ujian, kehadiran, penilaian kinerja, dan masih banyak lagi. SIS memungkinkan sekolah menyimpan berbagai jenis data dalam satu tempat sehingga memudahkan pemantauan perkembangan dan performa siswa [13].

Penerapan SIS menjadi bagian penting dari upaya digitalisasi administrasi sekolah, yang mentransformasi proses berbasis dokumen kertas menjadi sistem otomatis berbasis data. Penerapan SIS tidak hanya mempercepat proses administrasi sekolah, tetapi juga memperkuat manajemen berbasis data dan mendukung terciptanya lingkungan pendidikan yang transparan, efisien, serta adaptif terhadap transformasi digital [16]. Dengan sistem ini, kegiatan administratif yang sebelumnya memerlukan waktu lama seperti pencatatan nilai,

rekap absensi, dan pelaporan keuangan dapat dilakukan secara cepat dan akurat. Penerapan sistem informasi akademik terbukti meningkatkan kepuasan pengguna serta efektivitas proses administrasi di institusi pendidikan tinggi di Indonesia [14].

Modul-modul ini yang umum terdapat dalam sebuah SIS meliputi [15]:

1. Manajemen Pendaftaran (*Enrollment & Admissions*): pengelolaan proses penerimaan siswa baru dari formulir hingga verifikasi data.
2. Manajemen Akademik (*Academic Records*): meliputi pengelolaan rapor dan transkrip akademik.
3. Pelacakan Kehadiran (*Attendance Tracking*): otomasi absensi harian, pelaporan keterlambatan, dan *monitoring* kehadiran siswa.
4. Manajemen Keuangan (*Student Billing*): pengelolaan tagihan biaya sekolah seperti SPP, pelacakan status pembayaran.
5. Portal Pengguna (*Parent/Student Portal*): antarmuka daring yang memungkinkan siswa dan orang tua mengakses informasi akademik, absensi, dan pembayaran secara *real-time*.



Gambar 2.1 Diagram SIS [15]

Berdasarkan uraian tersebut, sistem informasi akademik menjadi komponen utama dalam penelitian ini karena digunakan untuk mendukung pengelolaan data akademik dan administrasi sekolah secara terpusat. Modul-modul yang terdapat dalam SIS juga menjadi acuan dalam perancangan fitur sistem yang dikembangkan, seperti pengelolaan data siswa, absensi, pembayaran, dan portal informasi pengguna. Dengan penerapan SIS, proses administrasi sekolah diharapkan menjadi lebih efektif, efisien, dan mudah diakses oleh seluruh pengguna sistem.

2.1.2 Digitalisasi Administrasi Sekolah

Salah satu bentuk transformasi digital yang krusial adalah digitalisasi administrasi, yakni proses mengintegrasikan teknologi digital dalam kegiatan administrasi sekolah seperti

pengelolaan data siswa, absensi, pelaporan kinerja guru, keuangan, dan komunikasi antar staf sekolah. Digitalisasi memiliki peran penting dalam meningkatkan kecepatan dan akurasi dalam pengelolaan data, sekaligus meminimalkan potensi kesalahan dalam proses administrasi. Dengan penerapan teknologi digital, sekolah dapat menciptakan tata kelola yang lebih adaptif terhadap perkembangan zaman [17].

Penerapan digitalisasi administrasi tidak hanya berdampak pada efisiensi kerja staf sekolah, tetapi juga mengubah cara berpikir, budaya kerja, dan pola manajemen di lingkungan sekolah. Transformasi digital menjadi sebuah kebutuhan untuk meningkatkan efisiensi dan efektivitas layanan pendidikan, baik dalam aspek administrasi maupun proses pembelajaran [18]. Dalam konteks pendidikan, digitalisasi memungkinkan guru dan staf administrasi untuk mengelola kehadiran, nilai, dan penjualan produk karya siswa, sementara siswa dan orang tua dapat memantau perkembangan akademik tanpa harus datang ke sekolah [19], [20], [21].

Berdasarkan teori yang telah dijelaskan, digitalisasi administrasi sekolah menjadi langkah penting dalam meningkatkan kualitas layanan pendidikan dan efisiensi pengelolaan data sekolah. Pada penelitian ini, konsep digitalisasi diterapkan melalui pengembangan sistem berbasis *web* yang mampu mengintegrasikan berbagai proses administrasi sekolah ke dalam satu platform. Dengan adanya digitalisasi, proses pengolahan data menjadi lebih cepat, transparan, dan mudah diakses oleh pihak sekolah maupun pengguna lainnya.

2.2 E-Commerce

E-Commerce (Electronic Commerce) adalah sistem perdagangan elektronik yang menggunakan jaringan internet dan teknologi digital untuk melakukan aktivitas jual-beli produk maupun jasa, serta pertukaran data dan pembayaran secara daring. Sistem ini tidak hanya mengubah bagaimana transaksi dilakukan, tetapi juga mengubah model bisnis, rantai pasokan, serta interaksi antara penjual, pembeli, dan platform digital. Riset terkini menunjukkan bahwa *e-commerce* mencakup beberapa dimensi utama seperti kenyamanan akses, jangkauan pasar global, efisiensi operasional, dan kemampuan untuk mengumpulkan data konsumen yang kemudian dapat dianalisis untuk strategi pemasaran dan layanan pelanggan [22].

Model transaksi dalam *e-commerce* dapat dibagi menjadi beberapa jenis: B2B (*Business to Business*), B2C (*Business to Consumer*), C2C (*Consumer to Consumer*), dan C2B (*Consumer to Business*). Dengan adanya model-model ini, bisnis dapat menjangkau konsumen secara langsung, antar konsumen, bahkan konsumen kepada bisnis. Fenomena ini

menunjukkan fleksibilitas struktur *e-commerce* dibanding perdagangan konvensional. Meski demikian, *e-commerce* juga menghadapi tantangan seperti isu keamanan data, kepercayaan konsumen, dan infrastruktur digital yang harus memadai agar sistem berjalan efektif. Meskipun *e-commerce* membuka banyak peluang, hambatan seperti keamanan, regulasi, logistik, dan infrastruktur digital tetap menjadi penghalang utama dalam pengembangannya [23].

Berdasarkan teori yang telah dijelaskan, *e-commerce* memiliki peran penting dalam mendukung proses transaksi digital yang lebih efektif, fleksibel, dan mudah dijangkau oleh pengguna. Pada penelitian ini, konsep *e-commerce* digunakan sebagai dasar pengembangan fitur penjualan produk karya siswa secara daring. Dengan adanya sistem *e-commerce*, sekolah tidak hanya dapat memasarkan produk siswa secara lebih luas, tetapi juga mendukung pengembangan keterampilan kewirausahaan digital dan pemanfaatan teknologi informasi dalam lingkungan pendidikan.

2.2.1 E-Commerce dalam Konteks Pendidikan

Penerapan *e-commerce* dalam lingkungan pendidikan bukan hanya terbatas pada aspek transaksi komersial, tetapi juga merupakan sarana strategis dalam pembelajaran kewirausahaan digital dan pemanfaatan hasil karya siswa sebagai produk nyata yang dapat dijual secara *online*. Pengalaman *e-commerce* dan literasi digital berpengaruh signifikan terhadap minat wirausaha digital siswa, dimana penggunaan platform daring memfasilitasi pemahaman siswa terhadap langkah-langkah bisnis digital seperti pemasaran, pengiriman, dan layanan pelanggan [24]. Institusi pendidikan yang mengintegrasikan *e-commerce* sebagai bagian dari kurikulum atau program kegiatan siswa berhasil meningkatkan kompetensi bisnis digital siswa dan memperkuat budaya kewirausahaan di institusi tersebut [25].

Dalam konteks institusi pendidikan yang memiliki program kewirausahaan, *e-commerce* memungkinkan siswa untuk menampilkan karya mereka secara daring, mengelola transaksi penjualan, berinteraksi dengan konsumen di luar lingkungan sekolah, dan belajar manajemen produksi hingga pemasaran dalam satu platform digital. Sistem ini akan memperluas pemanfaatan hasil karya siswa tidak hanya sebagai aktivitas pembelajaran, tetapi juga sebagai sumber pendapatan dan pengembangan karakter kewirausahaan. Penelitian lokal di Indonesia pun menunjukkan bahwa penggunaan platform *e-commerce* di kalangan siswa meningkatkan motivasi kreatif serta memperluas akses pasar produk karya siswa. Siswa yang aktif menggunakan *e-commerce* memiliki minat wirausaha digital yang

lebih tinggi dibanding yang tidak, dengan variabel literasi digital dan pengalaman *e-commerce* sebagai prediktor utama [26].

Berdasarkan uraian tersebut, penerapan *e-commerce* dalam lingkungan pendidikan dapat menjadi media pembelajaran kewirausahaan digital yang mendukung pengembangan kreativitas dan kemampuan bisnis siswa. Pada penelitian ini, fitur *e-commerce* digunakan untuk mendukung penjualan produk karya siswa secara *online* sehingga siswa tidak hanya belajar membuat produk, tetapi juga memahami proses pemasaran dan transaksi digital secara langsung. Dengan demikian, sistem yang dikembangkan diharapkan mampu meningkatkan pengalaman belajar sekaligus memperluas jangkauan pemasaran produk siswa.

2.2.2 Model dan Mekanisme Transaksi E-Commerce

Model transaksi pada *e-commerce* menjelaskan langkah-langkah yang dilalui mulai dari saat konsumen melakukan pemesanan hingga proses transaksi berakhir. Secara umum, proses transaksi dalam *e-commerce* terdiri dari beberapa tahapan utama, yaitu memilih produk, melakukan pemesanan atau *checkout*, melakukan pembayaran, memverifikasi transaksi, serta mengonfirmasi bahwa pesanan telah selesai diproses [27]. Setiap tahapan tersebut dibuat agar transaksi dapat berjalan secara terstruktur, jelas, dan dapat dipertanggungjawabkan.

Pada awalnya, konsumen memilih produk yang mereka inginkan lalu memasukkan produk tersebut ke dalam keranjang belanja. Setelahnya, sistem akan mencatat informasi pesanan termasuk jumlah barang, harga masing-masing produk, serta data pelanggan [28]. Langkah selanjutnya adalah proses pembayaran, yang bisa dilakukan dengan beberapa cara, antara lain melalui transfer bank, kartu kredit, dompet digital (*e-wallet*). Pemilihan metode pembayaran didasarkan pada kebutuhan sistem, tingkat keamanan yang diperlukan, dan keberadaan infrastruktur pendukung yang tersedia [29].

Dalam sistem dengan mekanisme transfer bank tanpa integrasi *payment gateway*, konsumen melakukan pembayaran secara mandiri melalui rekening tujuan, kemudian mengunggah bukti transfer sebagai bentuk konfirmasi pembayaran. Selanjutnya, administrator melakukan verifikasi terhadap bukti transfer tersebut sebelum status pesanan diperbarui menjadi “diproses” atau “selesai”. Mekanisme ini umum digunakan pada sistem *e-commerce* berskala kecil hingga menengah karena relatif mudah diterapkan dan tidak memerlukan integrasi langsung dengan layanan pembayaran pihak ketiga [30].

Selain itu, agar dapat mengelola pesanan dengan efisien dan memastikan ketersediaan stok, sistem *e-commerce* biasanya menerapkan batas waktu pembayaran. Jika dalam waktu tertentu konsumen belum melakukan pembayaran atau konfirmasi, pesanan dapat dibatalkan secara otomatis oleh sistem. Kebijakan ini bertujuan untuk mencegah terjadinya penumpukan pesanan yang belum selesai serta memastikan proses pengelolaan stok berjalan dengan efisien dan optimal [31]. Dengan adanya model dan mekanisme transaksi yang terstruktur, sistem *e-commerce* dapat memberikan pengalaman berbelanja yang lebih rapi, jelas, dan efisien bagi pengguna. Dalam konteks pendidikan, penerapan mekanisme ini juga bisa menjadi cara untuk siswa belajar dan memahami proses bisnis digital secara nyata serta terpadu [32].



Gambar 2.2 Proses *Checkout* dan Pembayaran [31]

Berdasarkan teori yang telah dijelaskan, model dan mekanisme transaksi *e-commerce* menjadi dasar penting dalam perancangan fitur transaksi pada sistem yang dikembangkan. Pada penelitian ini, mekanisme transaksi digunakan untuk mengatur proses pemesanan produk karya siswa mulai dari pemilihan produk, *checkout*, pembayaran, hingga verifikasi pesanan. Dengan mekanisme transaksi yang terstruktur, sistem diharapkan dapat membantu pengguna melakukan transaksi secara lebih mudah, aman, dan terorganisasi.

2.3 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak merupakan pendekatan sistematis yang digunakan untuk merancang, membangun, menguji, dan memelihara suatu sistem informasi agar sesuai dengan kebutuhan pengguna. Pemilihan metode pengembangan yang tepat sangat berpengaruh terhadap kualitas sistem, efisiensi proses pengembangan, serta kemampuan sistem dalam beradaptasi terhadap perubahan kebutuhan [33]. Dalam konteks

pengembangan sistem informasi berbasis *web* di lingkungan pendidikan, metode pengembangan yang fleksibel dan iteratif menjadi sangat penting karena kebutuhan pengguna dapat berkembang seiring dengan proses implementasi sistem [34], [35].

Pada penelitian ini, metode pengembangan perangkat lunak yang digunakan adalah pendekatan *Agile*, dengan kerangka kerja *Scrum* sebagai metode implementasinya. Metode ini dipilih karena mampu mendukung proses pengembangan sistem secara bertahap (*incremental*), memungkinkan adanya umpan balik dari pengguna secara berkelanjutan, serta meminimalkan risiko ketidaksesuaian sistem dengan kebutuhan pihak sekolah [36]. Dengan menggunakan metode *Agile Scrum*, setiap fitur sistem dapat dikembangkan, diuji, dan dievaluasi dalam siklus pengembangan yang singkat sehingga kualitas sistem dapat terus ditingkatkan secara berkelanjutan [37].

2.3.1 Metode Agile

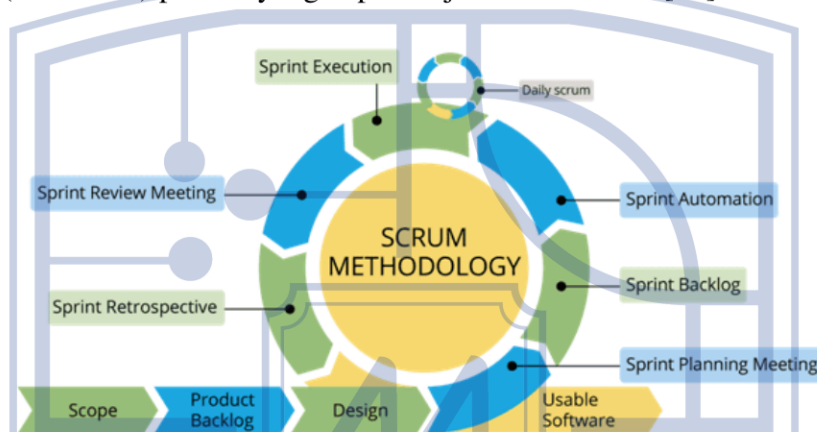
Metode *Agile* merupakan pendekatan pengembangan perangkat lunak yang menekankan fleksibilitas, kolaborasi tim, iterasi cepat, dan respons terhadap perubahan. Studi oleh “*Agile Software Development Projects–Unveiling the Human-Related Critical Success Factors*” menunjukkan bahwa dua faktor paling kuat dalam keberhasilan proyek *Agile* adalah kemampuan tim (*team capability*) dan keterlibatan pelanggan (*customer involvement*). Lebih lanjut, studi ini juga menemukan bahwa keamanan psikologis tim memiliki efek tidak langsung terhadap keberhasilan, melalui mediasi kemampuan tim dan keterlibatan pelanggan [38].

Studi “*Agile Practices and IT Development Team Well-Being: Unveiling the Path to Successful Project Delivery*” melaporkan bahwa praktik *Agile* yang baik juga berkorelasi positif dengan kesejahteraan tim, dan bahwa manajemen proyek *Agile* menuntut adaptasi tim dan keterlibatan pelanggan yang aktif [39]. Secara lebih spesifik, metode *Agile* memiliki karakteristik berikut [40]:

1. Iterasi pendek dan *incremental*: pekerjaan dibagi ke dalam unit kecil yang dikirim secara berkala.
2. Kolaborasi intensif antar anggota tim, serta keterlibatan *stakeholder* yang aktif [41].
3. Fokus pada pengiriman nilai lebih cepat dibanding dokumentasi yang berlebihan.
4. Adaptasi terhadap perubahan persyaratan bahkan di fase akhir pengembangan.

2.3.2 Metode Scrum

Scrum merupakan salah satu kerangka kerja (*framework*) dalam pendekatan *Agile* yang digunakan untuk mengelola dan mengembangkan perangkat lunak secara iteratif dan inkremental. *Scrum* berfokus pada kolaborasi tim, adaptasi terhadap perubahan, serta penyampaian produk secara bertahap melalui siklus pengembangan yang disebut *Sprint* [42]. Dalam *Scrum*, proses pengembangan dibagi ke dalam periode waktu tetap yang disebut *Sprint*, yang umumnya berlangsung antara 1 hingga 4 minggu. Setiap *Sprint* menghasilkan peningkatan (*increment*) produk yang dapat diuji dan dievaluasi [43].



Gambar 2.3 Tahapan Metode Scrum [44]

Scrum memiliki tiga peran utama, yaitu [43]:

1. *Product Owner*, bertanggung jawab dalam menentukan kebutuhan sistem dan mengelola *Product Backlog*. *Product Owner* memastikan fitur yang dikembangkan memiliki nilai dan prioritas yang jelas.
2. *Scrum Master*, berperan sebagai fasilitator yang memastikan proses *Scrum* berjalan sesuai aturan serta membantu tim mengatasi hambatan (*impediment*).
3. *Development Team*, bertugas merancang, mengembangkan, dan menguji sistem sesuai *backlog* yang telah ditentukan.

Selain peran, *Scrum* juga memiliki artefak utama, yaitu [43]:

1. *Product Backlog*: daftar seluruh kebutuhan sistem yang telah diprioritaskan.
2. *Sprint Backlog*: daftar pekerjaan yang dipilih untuk dikerjakan dalam *satu Sprint*.
3. *Increment*: hasil pengembangan pada akhir *Sprint* yang siap diuji.

Adapun tahapan kegiatan dalam *Scrum* meliputi [43], [44]:

1. *Product Backlog Creation*, penyusunan daftar kebutuhan sistem berdasarkan hasil analisis.
2. *Sprint Planning*, penentuan *backlog* mana yang akan dikerjakan dalam *satu Sprint*.
3. *Sprint Execution*, proses pengembangan sistem selama periode *Sprint*.

4. *Daily Scrum*, pertemuan harian singkat untuk memantau progres dan hambatan.
5. *Sprint Review*, evaluasi hasil pengembangan pada akhir *Sprint*.
6. *Sprint Retrospective*, evaluasi proses kerja tim untuk perbaikan pada *Sprint* berikutnya.

Berdasarkan teori yang telah dijelaskan, *Scrum* merupakan metode yang sesuai untuk mendukung pengembangan sistem secara iteratif dan terstruktur. Dalam penelitian ini, *Scrum* digunakan sebagai kerangka kerja pengembangan karena mampu membantu tim dalam mengelola proses pengembangan fitur secara bertahap melalui *Sprint*. Dengan penerapan *Scrum*, proses evaluasi, pengujian, dan perbaikan sistem dapat dilakukan secara berkelanjutan sehingga sistem yang dikembangkan lebih sesuai dengan kebutuhan pengguna dan lebih mudah dikontrol selama proses pengembangan.

2.4 Teknologi yang Digunakan dalam Pengembangan Sistem

Pengembangan aplikasi sistem informasi siswa dan penjualan produk karya siswa berbasis *web* membutuhkan dukungan teknologi yang tepat agar sistem dapat berjalan secara optimal, aman, dan mudah dikembangkan di masa mendatang [45]. Pemilihan teknologi dalam penelitian ini didasarkan pada kebutuhan sistem yang mencakup pengelolaan data akademik, transaksi penjualan, autentikasi pengguna, serta akses sistem secara daring oleh berbagai jenis pengguna, seperti administrator, guru, siswa, dan masyarakat umum [46].

Teknologi yang digunakan dalam pengembangan sistem ini mencakup platform berbasis *web* dengan arsitektur *client-server*, penerapan arsitektur layanan berbasis API, penggunaan *framework backend* untuk mengelola logika bisnis, *library frontend* untuk membangun antarmuka pengguna yang interaktif, serta sistem manajemen basis data relasional sebagai media penyimpanan data [47], [48]. Kombinasi teknologi tersebut diharapkan mampu menghasilkan sistem yang responsif, terstruktur, dan mudah dipelihara, sekaligus mendukung pengembangan sistem secara berkelanjutan sesuai dengan kebutuhan sekolah [49].

Berdasarkan teori yang telah dijelaskan, pemilihan teknologi dalam pengembangan sistem memiliki peranan penting dalam menentukan performa, keamanan, dan skalabilitas aplikasi. Pada penelitian ini, kombinasi teknologi berbasis *web* digunakan untuk mendukung pengelolaan data akademik dan transaksi penjualan produk siswa secara terintegrasi. Dengan penggunaan teknologi yang tepat, sistem diharapkan mampu memberikan pengalaman pengguna yang responsif, mudah diakses, serta mendukung pengembangan sistem di masa mendatang.

2.4.1 Website

Website adalah sistem perangkat lunak yang menggunakan arsitektur *client server* untuk mengirimkan konten dan fungsionalitas kepada pengguna melalui jaringan, seperti internet atau intranet. Dalam model ini, terdapat dua komponen utama yang berbeda [50]:

1. *Client-Side*: Merujuk pada *browser* pengguna. Komponen ini bertanggung jawab atas lapisan presentasi (*presentation layer*). *Browser* mengirimkan permintaan (*requests*) ke *server* dan kemudian memberikan respons untuk ditampilkan kepada pengguna.
2. *Server-Side*: Merujuk pada *server web* tempat logika aplikasi (*application layer*) dan data (*storage layer*) berada. Komponen ini bertanggung jawab untuk memproses permintaan klien, menjalankan logika bisnis (misalnya, memvalidasi *login*, menghitung data), berinteraksi dengan *database*, dan mengirimkan respons kembali ke klien.

Secara historis, model yang dominan adalah *Multi-Page Application* (MPA). Dalam arsitektur MPA, hampir semua logika *rendering* terjadi di sisi *server* (*Server-Side Rendering/SSR*). Setiap kali pengguna mengklik tautan atau mengirimkan formulir, *browser* akan membuat permintaan baru ke *server*, dan *server* merespons dengan mengirimkan halaman HTML yang sepenuhnya baru untuk dimuat ulang oleh *browser* [51]. Namun, untuk memenuhi tuntutan pengalaman pengguna yang lebih interaktif dan responsif, arsitektur modern telah bergeser ke model *Single Page Application* (SPA). Dalam arsitektur SPA, sebagian besar proses *rendering* dipindahkan ke sisi klien (*Client-Side Rendering/CSR*). Aplikasi memuat satu halaman HTML minimal pada awalnya, bersama dengan file JavaScript yang besar. Selanjutnya, JavaScript mengambil alih, menangani navigasi dan pembaruan UI secara dinamis tanpa memuat ulang seluruh halaman. Interaksi ini dilakukan dengan meminta dan menerima data (biasanya dalam format JSON) dari *server* [50], [51].

Arsitektur SPA ini secara natural mendorong adopsi arsitektur *decoupled* (terpisah) atau *headless*. Dalam model *headless*, *back-end* (*server*) tidak lagi bertanggung jawab untuk membuat HTML atau lapisan presentasi. Sebaliknya, *back-end* berfungsi murni sebagai penyedia data melalui API. *Frontend* bertindak sebagai konsumen independen yang mengonsumsi API tersebut [52]. Pemisahan ini memungkinkan fleksibilitas di mana satu *back-end* dapat melayani berbagai *frontend* (misalnya *website*, aplikasi seluler, dan IoT) secara bersamaan [53].

Berdasarkan teori yang telah dijelaskan, *website* menjadi media utama dalam penyampaian layanan sistem informasi secara daring kepada pengguna. Pada penelitian ini, *website* digunakan sebagai platform utama untuk mendukung akses informasi akademik dan transaksi penjualan produk karya siswa. Dengan pendekatan berbasis web, sistem dapat

diakses secara fleksibel melalui berbagai perangkat dan memudahkan interaksi antara pengguna dengan sistem secara *real-time*.

2.4.2 REST API

REST (*Representational State Transfer*) API merupakan gaya arsitektur yang digunakan untuk mengatur komunikasi data antar sistem melalui protokol HTTP. REST API memanfaatkan metode HTTP standar seperti *GET*, *POST*, *PUT*, dan *DELETE* untuk melakukan operasi terhadap sumber daya (*resource*) yang direpresentasikan dalam bentuk URI. Prinsip utama REST meliputi arsitektur *client-server*, *stateless communication*, serta penggunaan antarmuka yang seragam (*uniform interface*), sehingga sistem dapat dikembangkan secara terstruktur dan mudah dipelihara [54].

Dalam pengembangan aplikasi *web* modern, REST API umumnya digunakan pada arsitektur *decoupled* atau *headless*, di mana *backend* bertugas menyediakan data dan *frontend* berfungsi sebagai konsumen data secara terpisah. Namun, pada sistem yang dikembangkan dalam penelitian ini, REST API tidak diimplementasikan sebagai API publik yang berdiri sendiri, melainkan digunakan secara internal sebagai mekanisme pertukaran data antara lapisan *server* dan antarmuka pengguna [55].

Berdasarkan teori yang telah dijelaskan, REST API memiliki peranan penting dalam mendukung komunikasi data antar komponen sistem secara terstruktur dan efisien. Pada penelitian ini, REST API digunakan sebagai mekanisme pertukaran data antara *backend* dan *frontend* sehingga proses pengolahan serta pengiriman data dapat berjalan dengan lebih fleksibel dan terorganisasi. Dengan penggunaan REST API, sistem juga menjadi lebih mudah dikembangkan dan dipelihara di masa mendatang.

2.4.3 Framework Laravel

Laravel adalah salah satu *framework* PHP terbaik dalam meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan serta meningkatkan pengalaman bekerja dengan aplikasi untuk menyediakan sintaks yang ekspresif, jelas dan menghemat waktu. Keunggulan Laravel terletak pada struktur kode yang terorganisir, keamanan bawaan, serta dokumentasi yang komprehensif, hal ini yang menjadikan Laravel sebagai salah satu *framework* paling populer di dunia pengembangan *web* modern [56]. Sebagai sebuah *framework backend*, Laravel sangat ideal untuk membangun *server-side logic* dan REST API [57]. Fitur dari Laravel, seperti *routing*,

middleware, dan *Object-Relational Mapping*, memungkinkan pengembang untuk fokus pada logika bisnis alih-alih membangun infrastruktur aplikasi dari awal [58].

Berdasarkan teori yang telah dijelaskan, Laravel dipilih sebagai *framework backend* karena mampu mendukung pengembangan aplikasi *web* secara terstruktur, aman, dan efisien. Dalam penelitian ini, Laravel digunakan untuk mengelola logika bisnis, autentikasi pengguna, pengolahan data, dan integrasi sistem secara keseluruhan. Dengan fitur bawaan yang lengkap, Laravel dapat membantu mempercepat proses pengembangan sekaligus meningkatkan kualitas sistem yang dibangun.

2.4.4 Inertia.js

Inertia.js merupakan *library* yang berfungsi sebagai penghubung (*bridge*) antara *backend* Laravel dan *frontend* React tanpa memerlukan pembuatan REST API secara penuh [59]. Inertia.js memungkinkan pengembangan aplikasi *web* modern dengan konsep *Single Page Application* (SPA), namun tetap mempertahankan arsitektur *backend* tradisional berbasis *server-side* [59], [60]. Pada arsitektur ini, Laravel tetap berperan sebagai pengendali utama aplikasi, termasuk dalam hal *routing*, autentikasi, otorisasi, dan pengolahan data. Inertia.js bekerja dengan cara mengirimkan data dari *controller* Laravel ke komponen React dalam bentuk *props*, sehingga tidak diperlukan proses pemanggilan API secara terpisah menggunakan AJAX atau Axios untuk setiap halaman [59], [61]. Dengan menggunakan Inertia.js, sistem yang dikembangkan mampu menggabungkan keunggulan aplikasi berbasis SPA dan aplikasi *server-side* tradisional secara bersamaan, sehingga menghasilkan aplikasi yang responsif, aman, dan mudah dikembangkan [60].

Berdasarkan teori yang telah dijelaskan, Inertia.js memiliki peranan penting dalam menghubungkan *backend* Laravel dengan *frontend* React secara lebih sederhana dan efisien. Dalam penelitian ini, Inertia.js digunakan untuk mendukung pengembangan aplikasi berbasis SPA tanpa memerlukan pengelolaan REST API secara penuh. Dengan pendekatan tersebut, proses komunikasi data antara *backend* dan *frontend* menjadi lebih praktis serta mampu meningkatkan performa dan pengalaman pengguna aplikasi.

2.4.5 React.js (Frontend Library)

React.js adalah JavaScript *library* yang digunakan untuk membangun antarmuka pengguna (UI) yang dinamis dan responsif. ReactJS memiliki beberapa kelebihan seperti komponen yang *reusable*, *one-way data binding*, menggunakan *Virtual Document Object Model* (DOM), React Hooks, dan Komunitas Besar dan Ekosistem Luas. React

menggunakan konsep *component-based architecture*, dimana setiap elemen tampilan dikembangkan sebagai komponen terpisah, hal ini yang memungkinkan pengembang untuk menciptakan sistem yang mudah diperbarui, dan memiliki performa tinggi [62].

React.js menggunakan *Virtual Document Object Model* (DOM) yang mempercepat proses *rendering* elemen di *browser* sehingga aplikasi tetap responsif meskipun data yang ditampilkan ada dalam jumlah besar [63]. Fitur ini sangat penting untuk aplikasi *single-page* (SPA) yang melibatkan banyak komponen visual interaktif dan perubahan data yang dinamis. Dengan *state management* yang baik, React memudahkan sinkronisasi antara perubahan data di *backend* dengan tampilan antarmuka secara *real-time*, menciptakan pengalaman yang konsisten di seluruh perangkat [64].

Berdasarkan teori yang telah dijelaskan, React.js dipilih karena mampu membangun antarmuka pengguna yang interaktif, responsif, dan mudah dikembangkan. Pada penelitian ini, React.js digunakan untuk mendukung tampilan *frontend* sistem sehingga pengguna dapat berinteraksi dengan aplikasi secara lebih cepat dan nyaman. Dengan konsep *component-based architecture*, pengembangan antarmuka sistem menjadi lebih terstruktur dan mudah dipelihara.

2.4.6 TypeScript

TypeScript merupakan bahasa pemrograman yang merupakan *superset* dari JavaScript dan dikembangkan oleh Microsoft. TypeScript menambahkan fitur *static typing* ke dalam JavaScript, sehingga memungkinkan pendeteksian kesalahan kode sejak tahap pengembangan (*compile time*), bukan saat aplikasi dijalankan (*runtime*) [65]. Dalam pengembangan sistem ini, TypeScript digunakan pada sisi *frontend* bersama dengan React untuk meningkatkan keandalan, keterbacaan, dan skalabilitas kode. Dengan adanya penentuan tipe data secara eksplisit, pengembang dapat mengurangi risiko kesalahan akibat ketidaksesuaian tipe data, terutama pada aplikasi berskala menengah hingga besar [66].

Berdasarkan teori yang telah dijelaskan, TypeScript digunakan untuk meningkatkan kualitas dan keamanan kode program pada proses pengembangan aplikasi. Dalam penelitian ini, TypeScript membantu pengembang dalam mengurangi kesalahan penulisan kode serta meningkatkan keterbacaan dan pemeliharaan aplikasi *frontend*. Dengan penggunaan TypeScript, proses pengembangan sistem menjadi lebih terstruktur dan stabil.

2.4.7 MySQL (Database Management System)

MySQL adalah sistem manajemen basis data *open source* yang sangat fleksibel dan aman. MySQL menggunakan bahasa *Structured Query Language* (SQL) untuk mengelola data, mulai dari penyimpanan, pembaruan, pencarian, dan penghapusan data yang dilakukan secara efisien dan terstruktur. Salah satu keandalan MySQL adalah stabilitasnya dalam mengelola data berskala besar yang menjadi pilihan ideal bagi sistem pendidikan yang membutuhkan keandalan tinggi [67].

Dalam arsitektur *client-server*, MySQL berfungsi sebagai lapisan penyimpanan di sisi *server*. MySQL terkenal dengan keandalannya dan kinerjanya yang cepat untuk operasi baca/tulis, MySQL juga sangat kompatibel dengan berbagai bahasa pemrograman *server-side* serta *framework* seperti PHP dan Laravel. Hal ini menjadikannya fondasi yang umum digunakan untuk menyimpan dan mengelola data secara relasional [49].

Berdasarkan teori yang telah dijelaskan, MySQL digunakan sebagai sistem manajemen basis data karena mampu menyimpan dan mengelola data secara terstruktur, aman, dan efisien. Dalam penelitian ini, MySQL berperan sebagai media penyimpanan utama untuk data akademik, data pengguna, transaksi penjualan, dan informasi lainnya dalam sistem. Dengan penggunaan MySQL, pengelolaan data diharapkan menjadi lebih stabil dan mudah diintegrasikan dengan teknologi *backend* yang digunakan.

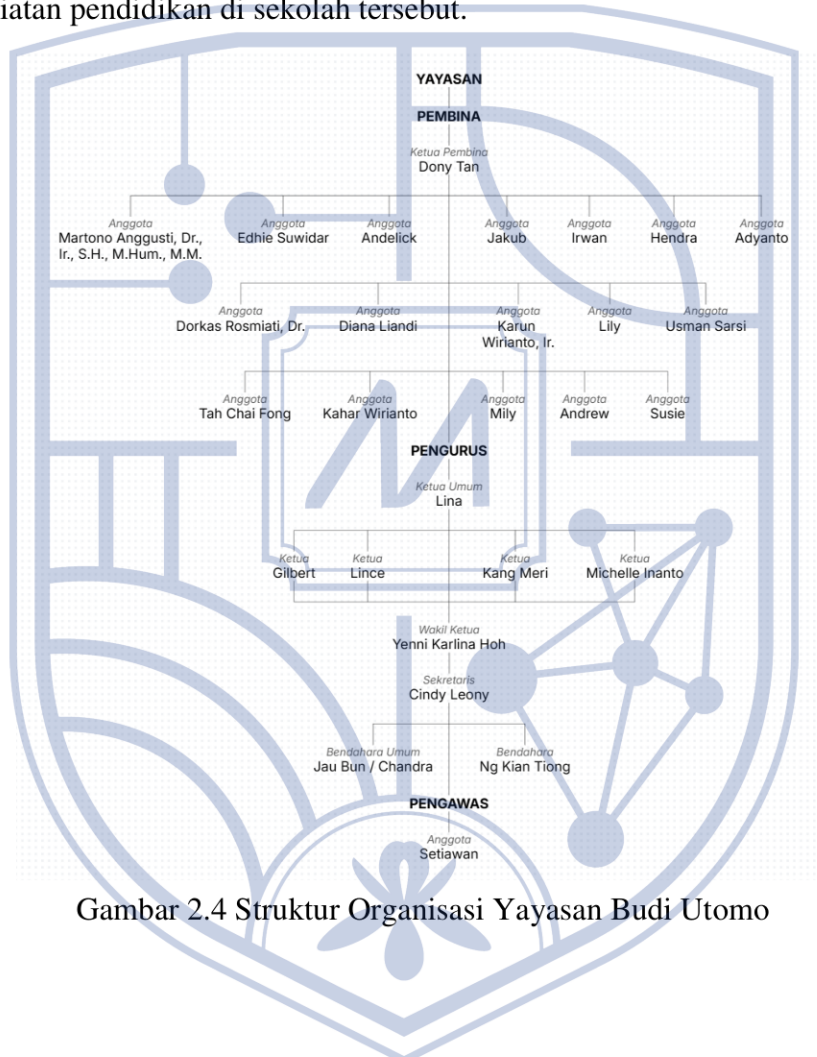
2.5 Tinjauan Objek Penelitian: Sekolah Swasta Budi Utomo Medan

Objek penelitian ini adalah Sekolah Swasta Budi Utomo Medan. Tinjauan ini akan memetakan proses bisnis yang berjalan, aktor yang terlibat, serta arsitektur sistem yang diusulkan yang bertujuan untuk memahami kebutuhan pengembangan sistem secara komprehensif.

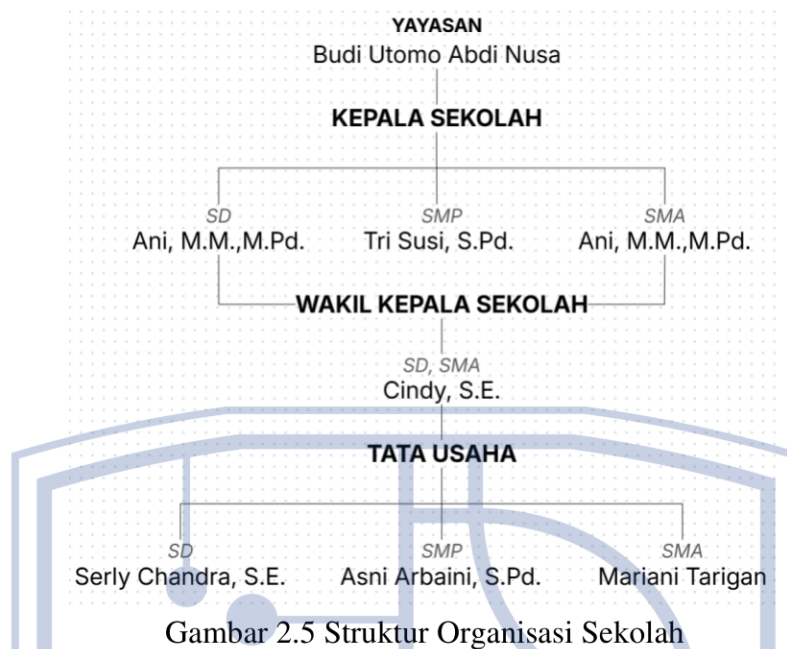
Berdasarkan tinjauan objek penelitian yang telah dilakukan, Sekolah Swasta Budi Utomo Medan memiliki kebutuhan terhadap sistem informasi terintegrasi yang mampu mendukung pengelolaan administrasi dan kegiatan akademik secara lebih efektif. Selain itu, pengembangan sistem juga diarahkan untuk mendukung program kewirausahaan siswa melalui fitur penjualan produk karya siswa secara digital. Dengan memahami kondisi dan kebutuhan sekolah, sistem yang dikembangkan diharapkan mampu memberikan solusi yang sesuai dengan proses bisnis yang berjalan di lingkungan sekolah.

2.5.1 Profil Singkat Sekolah Swasta Budi Utomo

Sekolah Swasta Budi Utomo merupakan lembaga pendidikan swasta yang berlokasi di Jalan Jurung No. 33, Kelurahan Pandau Hulu II, Kecamatan Medan Area, Kota Medan, Sumatera Utara [68]. Lembaga ini menyelenggarakan pendidikan secara komprehensif untuk beberapa jenjang, yang mencakup Taman Kanak-Kanak (TK), Sekolah Dasar (SD), Sekolah Menengah Pertama (SMP), dan Sekolah Menengah Atas (SMA). Sekolah Swasta Budi Utomo berada di bawah naungan sebuah yayasan yang mengelola dan mengawasi jalannya kegiatan pendidikan di sekolah tersebut.



Gambar 2.4 Struktur Organisasi Yayasan Budi Utomo



2.5.2 Proses Bisnis Saat Ini dan Identifikasi Aktor

Berdasarkan observasi yang telah diuraikan pada Bab 1, proses administrasi dan operasional di Sekolah Swasta Budi Utomo masih menggunakan sistem konvensional dan semi-komputerisasi (menggunakan Microsoft Excel). Mulai dari pengelolaan data siswa, absensi, nilai, dan pembayaran SPP belum terintegrasi. Selain itu, produk hasil karya siswa masih belum memiliki sarana penjualan secara digital.

Secara struktural, pengelolaan data dan administrasi sekolah berada di bawah tanggung jawab Kepala Sekolah. Sementara itu, proses akademik dikoordinasi oleh Kepala Sekolah dan dilaksanakan oleh para guru. Ada beberapa aktor kunci baik dari internal sekolah maupun eksternal yang akan terlibat dalam sistem yang diusulkan.

Aktor-aktor yang terlibat meliputi:

1. Administrator (Staf Tata Usaha)
 - a. Peran: Aktor internal dengan hak akses tertinggi yang bertanggung jawab dalam pengelolaan seluruh operasional dan data sistem.
 - b. Tugas Utama: Melakukan autentikasi (*login*) dan melihat *dashboard*, mengelola data master (siswa, guru, kelas, tahun ajaran, mata pelajaran, wali murid, dan ekstrakurikuler), mengelola absensi, nilai, dan rapor siswa, mengelola jenis biaya dan potongan siswa, mengelola pembayaran sekolah (SPP, uang pembangunan, dan biaya lainnya), mengelola transaksi penjualan produk karya siswa termasuk verifikasi pembayaran dan pembatalan pesanan, mengelola produk dan konten *website* (berita, pengumuman, dan pengaturan *landing page*), mengelola informasi

dan verifikasi pendaftaran siswa baru, mengelola pendaftaran ekstrakurikuler, mengelola jadwal, mengelola template serta melihat rekap evaluasi guru, mengelola pengguna dan hak akses, serta melihat laporan sistem.

2. Guru

- a. Peran: Aktor internal yang berfokus pada proses akademik dan pembinaan kewirausahaan siswa.
- b. Tugas Utama: Melakukan autentikasi (*login*) dan melihat *dashboard*, mengelola absensi siswa, menginput nilai siswa, melihat jadwal mengajar, mengelola produk karya siswa, serta melihat rekap siswa. Sebagai wali kelas, guru juga bertugas mengatur status kenaikan siswa, mencetak rapor siswa, dan mengelola catatan wali kelas.

3. Siswa

- a. Peran: Aktor internal sebagai pengguna utama yang mengakses informasi akademik dan non-akademik.
- b. Tugas Utama: Melakukan autentikasi (*login*) dan melihat *dashboard*, melihat absensi, nilai, jadwal pelajaran, dan rapor, memantau status pembayaran serta melakukan pembayaran sekolah (SPP dan biaya lainnya), melihat pengumuman sekolah, melihat dan mendaftar ekstrakurikuler serta memantau status pendaftarannya, mengisi evaluasi guru, serta melihat hasil evaluasi guru.

4. Pengunjung

- a. Peran: Aktor eksternal yang mengakses sistem tanpa autentikasi.
- b. Tugas Utama: Melihat halaman utama *website*, melihat informasi dan profil sekolah, membaca berita dan pengumuman, melakukan pendaftaran siswa baru, serta melihat katalog dan detail produk karya siswa tanpa dapat melakukan transaksi pembelian.

5. Customer

- a. Peran: Aktor eksternal yang menggunakan sistem *e-commerce* untuk melakukan transaksi pembelian produk karya siswa.
- b. Tugas Utama: Melakukan registrasi dan *login*, melihat katalog dan detail produk, menambahkan produk ke keranjang, melakukan pemesanan, mengunggah bukti pembayaran, melihat riwayat pesanan, serta mengelola profil akun.

2.5.3 Tinjauan Studi Kasus: Penjualan Produk Karya Siswa

Salah satu fokus utama dari penelitian ini adalah mendukung program kewirausahaan di lingkungan sekolah. Istilah “Produk Karya Siswa” merujuk pada hasil kreativitas siswa yang dihasilkan melalui proses pembelajaran maupun kegiatan ekstrakurikuler. Produk tersebut dapat berupa kerajinan tangan, olahan makanan, maupun karya desain grafis yang memiliki nilai jual.

Sistem yang dikembangkan menyediakan platform *e-commerce* sederhana dengan mekanisme sebagai berikut:

1. **Pengelolaan Produk:** Administrator dapat menambahkan, mengubah, dan menghapus data produk karya siswa yang ditampilkan dalam sistem.
2. **Akses Produk:** Pengunjung dapat melihat katalog dan detail produk, sedangkan *customer* dapat melakukan pemesanan setelah melakukan autentikasi.
3. **Proses Pemesanan:** *Customer* dapat memilih produk, menambahkan produk ke keranjang, dan melakukan proses pemesanan (*checkout*) melalui sistem.
4. **Metode Pembayaran:** Sistem menggunakan metode pembayaran nontunai melalui transfer bank. *Customer* diwajibkan untuk mengunggah bukti pembayaran sebagai bagian dari proses verifikasi.
5. **Validasi Pesanan:** Administrator bertugas melakukan verifikasi terhadap bukti pembayaran yang diunggah oleh customer serta mengelola status pesanan. Sistem akan secara otomatis membatalkan pesanan apabila dalam waktu 24 jam setelah pemesanan dilakukan tidak terdapat konfirmasi pembayaran.

Berdasarkan studi kasus yang telah dijelaskan, penjualan produk karya siswa melalui platform digital dapat menjadi sarana pendukung pembelajaran kewirausahaan dan pengembangan kreativitas siswa. Pada penelitian ini, sistem *e-commerce* digunakan untuk membantu proses promosi dan transaksi produk karya siswa secara lebih terstruktur dan mudah diakses. Dengan adanya sistem tersebut, sekolah diharapkan dapat memperluas jangkauan pemasaran produk sekaligus memberikan pengalaman bisnis digital secara langsung kepada siswa.

2.5.4 Arsitektur Sistem Informasi

Untuk memenuhi kebutuhan seluruh aktor dan memisahkan fungsi publik dengan fungsi internal, arsitektur sistem yang dikembangkan akan dibagi menjadi empat (4) *subdomain* utama:

1. *Website* Utama (Domain Publik): Berfungsi sebagai halaman utama (*landing page*) yang menampilkan profil sekolah, berita umum, dan informasi kontak. *Website* ini menjadi pintu akses utama ke sistem serta menyediakan tautan menuju *subdomain* lainnya.
2. *Website* Pendaftaran Siswa Baru: Merupakan portal khusus yang digunakan oleh pengunjung untuk melakukan pendaftaran siswa baru secara daring melalui sistem.
3. *Website* Penjualan Karya Siswa: Merupakan platform *e-commerce* yang menampilkan produk karya siswa. Pengunjung dapat melihat katalog dan detail produk, sedangkan *customer* dapat melakukan transaksi pembelian setelah melakukan *login*.
4. *Website* Internal (Sistem Informasi Sekolah): Merupakan sistem terintegrasi yang memerlukan autentikasi (*login*) dan digunakan oleh administrator, guru, dan siswa. Sistem ini menyediakan seluruh fitur pengelolaan data akademik dan administrasi sekolah.

Berdasarkan kebutuhan sistem yang telah dianalisis, pembagian arsitektur ke dalam beberapa *subdomain* dilakukan untuk memisahkan fungsi publik dan fungsi internal agar pengelolaan sistem menjadi lebih terstruktur dan aman. Pada penelitian ini, setiap *subdomain* memiliki peran yang berbeda sesuai kebutuhan pengguna, mulai dari penyampaian informasi publik, pendaftaran siswa baru, transaksi penjualan produk siswa, hingga pengelolaan administrasi akademik internal sekolah. Dengan arsitektur tersebut, sistem diharapkan mampu memberikan performa dan pengelolaan layanan yang lebih optimal.

2.6 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa standar yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak berbasis objek (*Object-Oriented Programming*) [69]. UML bukan sekadar kumpulan diagram, melainkan sebuah metodologi untuk memahami struktur dan perilaku sistem secara mendalam [70]. Dalam pengembangan aplikasi berbasis Laravel dan React pada Sekolah Budi Utomo ini, UML berfungsi sebagai jembatan untuk mentransformasikan kebutuhan bisnis sekolah ke dalam desain teknis yang sistematis.

Berdasarkan teori yang telah dijelaskan, UML memiliki peranan penting dalam membantu proses analisis dan perancangan sistem secara terstruktur. Pada penelitian ini, UML digunakan sebagai alat pemodelan untuk menggambarkan kebutuhan sistem, alur proses, serta interaksi antar pengguna dengan sistem yang dikembangkan. Dengan

penggunaan UML, proses perancangan sistem menjadi lebih mudah dipahami dan dapat membantu pengembang dalam mengimplementasikan sistem sesuai kebutuhan pengguna.

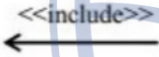
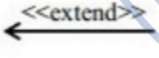
2.6.1 Use Case Diagram

Use Case Diagram merupakan diagram UML yang digunakan untuk merepresentasikan pemodelan perilaku (*behavior*) sistem informasi yang akan dibangun [70]. Diagram ini menggambarkan interaksi antara aktor eksternal dengan sistem melalui fitur-fitur yang disediakan [71]. Tujuan utama *use case diagram* adalah untuk mengidentifikasi dan mendeskripsikan fungsionalitas sistem dari perspektif pengguna (*user-oriented*) [71].

Notasi dan Aturan Semantik *Use Case*:

1. *Actor* merupakan entitas eksternal yang berinteraksi dengan sistem. Aktor dapat berupa manusia, perangkat keras, maupun sistem lain yang berada di luar batas sistem. Aktor merepresentasikan peran (*role*), bukan individu tertentu [71]. Dalam penelitian ini, aktor yang terlibat meliputi, administrator, guru, siswa, calon siswa/masyarakat umum.
2. *Use Case* adalah representasi dari fungsi atau layanan yang disediakan oleh sistem kepada aktor. *Use case* menggambarkan aktivitas atau proses yang dapat dilakukan oleh aktor dalam sistem [71]. Penamaan *use case* harus menggunakan kata kerja yang menggambarkan aksi, seperti, mengelola data siswa, melakukan pendaftaran, melihat informasi sekolah.
3. *Relationship* (Relasi) [70]:
 - a. *Association* merupakan hubungan komunikasi antara aktor dengan *use case*. Relasi ini menunjukkan bahwa aktor dapat berinteraksi atau menggunakan fungsi tersebut.
 - b. *Generalization* menunjukkan hubungan pewarisan (*inheritance*). Relasi ini dapat terjadi antara aktor dengan aktor (aktor khusus mewarisi hak akses aktor umum) atau antara *use case* dengan *use case* (*use case* spesifik merupakan turunan dari *use case* yang lebih umum).
 - c. *Include* (<<include>>) menunjukkan bahwa suatu *use case* selalu menyertakan *use case* lain sebagai bagian dari alur prosesnya (bersifat wajib). Jika *use case* A memiliki relasi *include* ke *use case* B, maka setiap kali A dijalankan, B juga akan dijalankan sebagai bagian dari proses tersebut.
 - d. *Extend* (<<extend>>) menunjukkan bahwa suatu *use case* dapat diperluas oleh *use case* lain pada kondisi tertentu (bersifat opsional).

Tabel 2.1 Simbol *Use Case Diagram* [72]

Simbol	Nama	Keterangan
	<i>Actor</i>	Entitas eksternal yang berinteraksi dengan sistem. Dapat berupa manusia, perangkat keras, atau sistem lain.
	<i>Use Case</i>	Representasi fungsi atau layanan yang disediakan sistem kepada aktor.
	<i>Association</i>	Hubungan komunikasi antara aktor dan <i>use case</i> yang menunjukkan adanya interaksi.
	<i>Generalization</i>	Hubungan pewarisan antara aktor dengan aktor atau <i>use case</i> dengan <i>use case</i> yang menunjukkan spesialisasi.
	<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> selalu menyertakan <i>use case</i> lain sebagai bagian dari prosesnya.
	<i>Extend</i>	Menunjukkan perluasan fungsi dari <i>use case</i> utama yang dijalankan pada kondisi tertentu.

Berdasarkan teori yang telah dijelaskan, *Use Case Diagram* digunakan untuk menggambarkan interaksi antara pengguna dengan sistem yang dikembangkan. Dalam penelitian ini, *Use Case Diagram* membantu mengidentifikasi kebutuhan fungsional sistem berdasarkan peran masing-masing aktor, seperti administrator, guru, siswa, pengunjung, dan customer. Dengan adanya diagram ini, proses analisis kebutuhan sistem menjadi lebih terstruktur dan mudah dipahami sebelum tahap implementasi dilakukan.

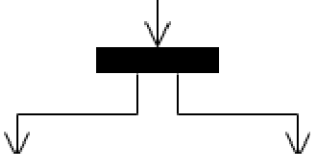
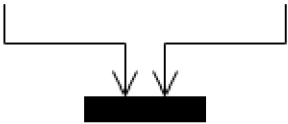
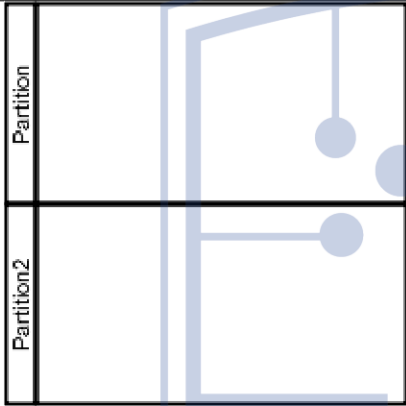
2.6.2 Activity Diagram

Activity Diagram menggambarkan aliran prosedural dari satu aktivitas ke aktivitas lainnya dalam sistem. Jika *Use Case* berfokus pada "apa" yang dilakukan, *Activity Diagram* berfokus pada "bagaimana" urutan logika itu berjalan secara internal. Diagram ini sangat efektif untuk memodelkan proses bisnis sekolah yang memiliki banyak percabangan keputusan (*decisions*) dan proses paralel [72].

Simbol-simbol yang digunakan dalam Activity Diagram beserta fungsinya dijelaskan pada Tabel 2.2:

Tabel 2.2 Simbol Activity Diagram [72]

Simbol	Nama	Keterangan
	Initial Node	Menandakan titik awal dimulainya suatu alur aktivitas.
	Action	Menggambarkan suatu aktivitas atau tindakan yang dijalankan dalam alur proses.
	Control Flow	Menunjukkan perpindahan kontrol/urutan dari satu aktivitas ke aktivitas berikutnya.
	Decision Node	Menunjukkan pemilihan jalur berdasarkan kondisi tertentu; satu alur masuk dan beberapa alur keluar dengan penjaga kondisi.
	Merge Node	Menyatukan kembali beberapa alur alternatif menjadi satu alur, beberapa alur masuk dan satu alur keluar, tanpa kondisi.

	Fork Node	Memecah satu alur menjadi beberapa alur yang berjalan secara bersamaan (paralel).
	Join Node	Menyatukan kembali alur-alur paralel; alur berlanjut setelah seluruh cabang selesai (sinkronisasi).
	Swimlane and Partition	Memisahkan aktivitas berdasarkan aktor atau penanggung jawabnya.
	Activity Final	Menandakan berakhirnya seluruh alur aktivitas dalam diagram.

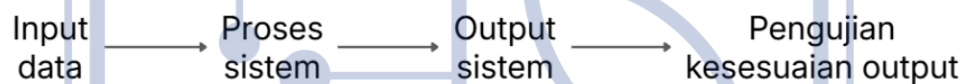
Berdasarkan teori yang telah dijelaskan, *Activity Diagram* digunakan untuk menggambarkan alur proses dan logika aktivitas dalam sistem secara lebih detail. Pada penelitian ini, *Activity Diagram* membantu memvisualisasikan proses bisnis seperti pengelolaan data akademik, proses pembayaran, dan transaksi penjualan produk karya siswa. Dengan adanya diagram ini, urutan aktivitas dan interaksi antar aktor dalam sistem dapat dipahami dengan lebih jelas sebelum sistem diimplementasikan.

2.7 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan proses untuk mengevaluasi dan memverifikasi bahwa sistem yang dikembangkan telah berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Pengujian dilakukan untuk menemukan kesalahan (*error*), memastikan setiap fungsi berjalan dengan baik, serta menjamin kualitas sistem sebelum digunakan oleh pengguna akhir [73]. Tujuan utama pengujian perangkat lunak adalah untuk memastikan bahwa sistem memenuhi kebutuhan fungsional, meminimalkan kesalahan, serta meningkatkan keandalan dan kualitas perangkat lunak [74].

Black Box Testing adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsi sistem tanpa melihat struktur kode program di dalamnya. Pengujian dilakukan berdasarkan *input* dan *output* yang dihasilkan sistem, dengan tujuan untuk memastikan bahwa setiap fitur berjalan sesuai dengan spesifikasi yang telah dirancang [75]. Dalam metode ini, penguji hanya memeriksa apakah sistem [76], [77]:

1. Menerima *input* dengan benar.
2. Memproses data sesuai logika yang dirancang.
3. Menghasilkan *output* yang sesuai.
4. Menampilkan pesan kesalahan jika terjadi kesalahan *input*.



Gambar 2.6 Proses *Black Box Testing* [75]

Selain pengujian fungsional, kualitas sebuah sistem juga perlu dinilai dari sudut pandang pengguna yang menggunakannya secara langsung. Pengujian berbasis pengguna merupakan metode pengujian yang melibatkan pengguna akhir untuk mencoba fitur-fitur sistem dan memberikan penilaian terhadap kesesuaian sistem dengan kebutuhan mereka. Berbeda dengan *Black Box Testing* yang menilai kebenaran keluaran terhadap masukan, pengujian berbasis pengguna menekankan aspek penerimaan dan pengalaman pengguna, seperti kemudahan penggunaan, kejelasan tampilan antarmuka, serta kesesuaian fitur dengan kebutuhan pengguna [78]. Penilaian dari pengguna umumnya dikumpulkan melalui kuesioner sehingga hasilnya dapat dijadikan bahan evaluasi untuk memastikan bahwa sistem yang dikembangkan benar-benar dapat digunakan dengan baik oleh penggunanya.

Untuk mengukur tanggapan pengguna pada kuesioner, penelitian ini menggunakan skala Likert. Skala Likert adalah skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang terhadap suatu objek atau fenomena tertentu [79]. Setiap pernyataan dalam kuesioner diberi pilihan jawaban berjenjang, yaitu 1 (Sangat Tidak Setuju), 2 (Tidak Setuju), 3 (Netral), 4 (Setuju), dan 5 (Sangat Setuju). Jawaban responden kemudian diberi skor, dijumlahkan, lalu dihitung persentasenya terhadap skor maksimal dan diinterpretasikan ke dalam kategori tertentu, misalnya sangat baik, baik, cukup, atau kurang. Dengan demikian, skala Likert memungkinkan penilaian pengguna yang bersifat kualitatif diubah menjadi data kuantitatif yang dapat dianalisis secara objektif [80]. Nilai persentase yang diperoleh selanjutnya diinterpretasikan ke dalam kategori penerimaan pengguna

berdasarkan rentang persentase tertentu. Kriteria interpretasi skor yang digunakan dalam penelitian ini ditunjukkan pada Tabel 2.3.

Tabel 2.3 Kriteria Interpretasi Skor [81]

Persentase	Kategori
0% – 20%	Sangat Tidak Baik
21% – 40%	Tidak Baik
41% – 60%	Cukup
61% – 80%	Baik
81% – 100%	Sangat Baik

Berdasarkan teori yang telah dijelaskan, pengujian perangkat lunak memiliki peranan penting dalam memastikan kualitas dan keandalan sistem sebelum digunakan oleh pengguna. Pada penelitian ini, metode *Black Box Testing* digunakan untuk menguji fungsi-fungsi sistem berdasarkan *input* dan *output* yang dihasilkan, sedangkan pengujian berbasis pengguna digunakan untuk mengetahui tingkat kepuasan dan kemudahan penggunaan sistem. Dengan kombinasi kedua metode tersebut, sistem diharapkan dapat berjalan sesuai kebutuhan pengguna dan memiliki kualitas yang baik.