

## BAB II. TINJAUAN PUSTAKA

### 2.1. Sistem Informasi

Sistem informasi sekarang sudah jadi bagian tak terpisahkan dari banyak bidang, salah satunya pendidikan. Kalau digunakan dengan baik, sistem ini bisa bantu organisasi atur data dan informasi lebih rapi serta hemat waktu. Khusus di dunia pendidikan, fungsinya tidak cuma buat nyimpan dan ambil data, tapi juga untuk berbagai urusan akademik dan administrasi sehari-hari (Nurman Hidayat & Kusuma Hati, 2021).

Secara sederhana, sistem informasi merupakan sekumpulan komponen yang terintegrasi, yang memiliki fungsi untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi, dengan tujuan mendukung proses pengambilan keputusan, koordinasi, pemantauan, analisis, serta visualisasi dalam suatu organisasi. Menurut (Ghosh et al., 2022), sistem informasi dapat dipahami sebagai desain khusus yang bertujuan untuk menyediakan informasi yang mendukung aktivitas operasional, manajemen, dan proses pengambilan keputusan di dalam suatu organisasi.

Sistem informasi memainkan peran penting dalam meningkatkan kualitas pengambilan keputusan. Dengan ketersediaan data yang akurat dan terkini, manajer lembaga pendidikan dapat mengambil keputusan yang lebih terinformasi berdasarkan analisis data yang komprehensif. Misalnya, informasi tentang kehadiran siswa, prestasi akademik, dan aspek keuangan dapat digunakan untuk merancang strategi guna meningkatkan kualitas pendidikan dan efisiensi operasional lembaga.

Sistem informasi membuat segalanya lebih terbuka dan bisa dipertanggungjawabkan. Setiap kegiatan dan transaksi yang berjalan di dalamnya bisa dilacak serta diperiksa dengan teliti, jadi risiko kesalahan atau kecurangan bisa dikurangi. Ini sangat penting banget di dunia pendidikan, di mana integritas dan kepercayaan jadi hal yang super krusial.

Sistem informasi juga memungkinkan automasi berbagai proses yang sebelumnya dilakukan secara manual. Proses seperti mendaftarkan siswa, mengatur jadwal, dan menghitung nilai bisa dikerjakan jauh lebih cepat dan tepat kalau pakai sistem informasi. Otomatisasi seperti ini tidak cuma bikin hemat waktu dan tenaga, tapi juga membantu mengurangi risiko salah yang sering disebabkan manusia.

Sistem informasi sangat membantu dalam mengambil keputusan yang lebih akurat. Dengan data yang akurat dan selalu *up-to-date*, para pengelola sekolah atau universitas bisa bikin pilihan yang lebih tepat lewat analisis data yang mendalam. Seperti info soal absensi siswa, prestasi akademik, dan kondisi

keuangan bisa digunakan untuk rencana strategis, naikin kualitas pendidikan atau bikin operasional lebih efisien.

Namun, masih banyak lembaga pendidikan yang menghadapi kesulitan dalam mengelola data yang kompleks dan beragam, karena prosesnya masih dilakukan secara manual. Oleh karena itu, diperlukan pengembangan sistem informasi akademik yang mampu mengintegrasikan semua aspek administrasi dan akademik secara terpadu, sehingga masalah-masalah tersebut dapat ditangani dengan lebih efektif dan efisien.

## **2.2. Sistem Akademik**

Sistem akademik merupakan fondasi utama dalam pengelolaan dan manajemen pendidikan di sebuah lembaga. Sistem ini mencakup berbagai komponen yang saling berhubungan, yang dirancang untuk mengelola seluruh aspek akademik mulai dari penerimaan siswa, pengelolaan kurikulum, penjadwalan kelas, hingga evaluasi dan penilaian. Dalam era digital seperti saat ini, pengembangan dan implementasi sistem akademik yang berbasis teknologi informasi menjadi sangat penting untuk meningkatkan efisiensi, efektivitas, dan kualitas layanan pendidikan (Marsehan et al., 2025).

Sistem akademik modern biasanya terdiri dari beberapa modul yang dirancang untuk mengelola berbagai aspek penting dalam proses pendidikan. Modul-modul ini mencakup penerimaan siswa baru, manajemen data siswa, manajemen kurikulum dan mata pelajaran, penjadwalan kelas, pengelolaan nilai dan rapor, serta pelaporan dan analisis data. Dengan adanya sistem akademik yang terintegrasi, semua data yang terkait dengan kegiatan akademik dapat dikelola dengan lebih sistematis dan mudah diakses oleh pihak yang berwenang (Hariyanto et al., 2021).

Modul penerimaan siswa baru merupakan salah satu komponen penting dalam sistem akademik. Modul ini membantu lembaga pendidikan dalam mengelola proses pendaftaran siswa mulai dari pengisian formulir, verifikasi dokumen, hingga pengumuman hasil seleksi. Dengan adanya modul ini, proses penerimaan siswa baru dapat dilakukan dengan lebih cepat, transparan, dan akurat. Selain itu, data calon siswa yang masuk ke dalam sistem dapat disimpan dan diolah untuk berbagai keperluan, seperti analisis demografi atau perencanaan kapasitas kelas (Beatriks et al., 2025).

Modul manajemen data siswa berfungsi untuk menyimpan dan mengelola seluruh informasi yang terkait dengan siswa, termasuk data pribadi, riwayat pendidikan, dan catatan prestasi akademik. Melalui modul ini, pihak sekolah dapat dengan mudah mengakses informasi siswa yang diperlukan untuk berbagai keperluan, seperti pembuatan laporan, pemantauan perkembangan akademik, dan komunikasi dengan orang tua. Selain itu, modul ini memungkinkan

penyimpanan data secara aman dan terstruktur, sehingga dapat meminimalkan risiko kehilangan maupun kerusakan data (Dwiyatno et al., 2022).

Modul pengelolaan nilai merupakan salah satu komponen penting dalam sistem akademik. Melalui modul ini, guru dapat dengan mudah melakukan input, pengelolaan, serta perhitungan nilai siswa secara otomatis. Selain itu, modul ini juga mampu menghasilkan laporan hasil belajar atau rapor yang berisi informasi mengenai prestasi akademik serta catatan kehadiran siswa. Kehadiran modul ini menjadikan proses penilaian lebih cepat, akurat, dan mengurangi kemungkinan terjadinya kesalahan manusia. Data nilai yang tersimpan pun menjadi lebih aman serta dapat diakses kapan pun oleh pihak yang memiliki otoritas (Beatriks et al., 2025).

Sementara itu, modul keuangan berfungsi untuk mendukung pengelolaan informasi keuangan agar pencatatan dapat dilakukan secara sistematis dan akurat, sehingga risiko ketidakterkendalian data keuangan dapat diminimalkan. Modul ini mempermudah penyusunan laporan keuangan dalam waktu singkat, mengurangi potensi kesalahan input maupun perhitungan, serta meningkatkan efisiensi kerja melalui proses otomatisasi sistem (Ahmad Khainur Nadhif et al., 2024).

Untuk membangun sistem akademik dengan modul-modul yang ada maka dibutuhkan panduan pendekatan sistematis dan terstruktur untuk menciptakan perangkat lunak berkualitas tinggi yaitu *System Development Life Cycle* (SDLC).

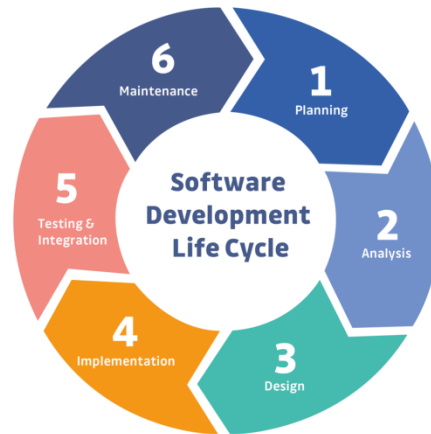
### **2.3. *System Development Life Cycle* (SDLC)**

Siklus Hidup Pengembangan Sistem, atau *System Development Life Cycle* (SDLC), adalah kerangka kerja yang digunakan untuk memahami dan mengelola seluruh tahap dalam proses pengembangan perangkat lunak, mulai dari perencanaan hingga implementasi. SDLC berperan penting dalam membantu tim pengembang untuk merancang, membangun, menguji, dan mengimplementasikan sistem yang sesuai dengan kebutuhan pengguna secara terstruktur, sistematis, dan efisien (Petrovic et al., 2020).

Model SDLC membentuk tulang punggung rekayasa perangkat lunak praktik rekayasa perangkat lunak, memandu pendekatan sistematis dan terstruktur untuk menciptakan produk perangkat lunak berkualitas tinggi. Seiring perkembangan teknologi berkembang dan permintaan pasar menjadi lebih dinamis, organisasi pengembangan perangkat lunak menghadapi tantangan untuk memilih model SDLC yang paling tepat untuk memenuhi persyaratan proyek secara efisien dan efektif. Untuk membuat keputusan yang tepat, pengembang, manajer proyek, dan insinyur kualitas membutuhkan pemahaman yang

komprehensif tentang keuntungan, kekurangan, dan kesesuaian aplikasi dari berbagai model SDLC dalam konteks rekayasa kualitas perangkat lunak (Pargaonkar, 2023).

SDLC mencakup serangkaian tahap, masing-masing memiliki tujuan spesifik dan aktivitas tertentu yang harus diselesaikan sebelum melanjutkan ke tahap berikutnya.



**Gambar 2.1. SDLC Proses**

Sumber: (Pargaonkar, 2023).

Tahap *Planning* (Perencanaan) merupakan langkah awal dalam *System Development Life Cycle* (SDLC) yang berfokus pada identifikasi masalah atau peluang, penetapan tujuan proyek, serta penentuan sumber daya yang dibutuhkan. Pada tahap ini, biasanya dilakukan feasibility study atau studi kelayakan untuk menilai kelayakan proyek dari aspek teknis, ekonomi, dan operasional. Studi tersebut berfungsi untuk memastikan bahwa sumber daya yang tersedia dapat dimanfaatkan secara optimal guna mencapai tujuan proyek.

Tahap *Analysis* (Analisis Kebutuhan) berfokus pada pengumpulan informasi dari berbagai pemangku kepentingan untuk memahami kebutuhan dan ekspektasi mereka terhadap sistem yang akan dibangun. Pada tahap ini, analis sistem bekerja sama dengan pengguna akhir dalam mengidentifikasi fitur dan fungsi yang diperlukan, kemudian mendokumentasikannya dalam bentuk spesifikasi kebutuhan. Dokumen spesifikasi ini menjadi acuan utama bagi tahap desain selanjutnya.

Tahap *Design* (Perancangan) mencakup pengembangan arsitektur sistem serta perancangan rinci dari komponen-komponen utama, seperti struktur data, antarmuka pengguna, dan alur kerja sistem. Tim pengembang pada tahap ini membuat blueprint atau rancangan dasar yang menjadi pedoman untuk proses pengembangan. Desain yang baik perlu memperhatikan aspek skalabilitas,

keamanan, efisiensi, dan kemudahan pemeliharaan agar sistem dapat berfungsi dengan optimal dalam jangka panjang.

Tahap *Implementation* (Implementasi) merupakan proses penerapan sistem ke dalam lingkungan operasional. Pada tahap ini, data dari sistem lama dimigrasikan ke sistem baru, dan pengguna diberikan pelatihan untuk menggunakan sistem tersebut. Implementasi dapat dilakukan secara bertahap maupun sekaligus, tergantung pada tingkat kompleksitas dan risiko proyek. Dukungan teknis yang memadai sangat diperlukan agar proses transisi berjalan lancar serta untuk mengatasi potensi kendala yang muncul.

Tahap *Testing* (Pengujian) bertujuan untuk memastikan bahwa sistem beroperasi sesuai dengan persyaratan yang ditetapkan dan bebas dari kesalahan atau bug. Pengujian dilakukan pada berbagai tingkatan, termasuk pengujian unit, pengujian integrasi, pengujian sistem, dan *user acceptance testing* (UAT). Setiap jenis pengujian memiliki fokus yang berbeda, tetapi secara keseluruhan bertujuan untuk mendeteksi dan memperbaiki kesalahan sebelum sistem digunakan secara resmi. Pengujian yang komprehensif memastikan kualitas dan keandalan system,

Fase *Maintenance* (Pemeliharaan) adalah fase terakhir dalam SDLC, di mana sistem yang telah diimplementasikan terus dipantau dan disempurnakan agar tetap relevan dengan kebutuhan organisasi. Aktivitas pemeliharaan meliputi perbaikan kesalahan, pembaruan keamanan, penambahan fitur baru, dan penyesuaian terhadap perubahan lingkungan operasional. Pemeliharaan berkelanjutan memastikan sistem terus berfungsi secara optimal dan mendukung kinerja organisasi dalam jangka panjang.

Salah satu model SDLC yang cocok untuk pengembangan sistem informasi akademik adalah *Rapid Application Development* (RAD), karena metode ini memungkinkan proses pengembangan sistem yang lebih cepat dan implementasi yang lebih mudah.

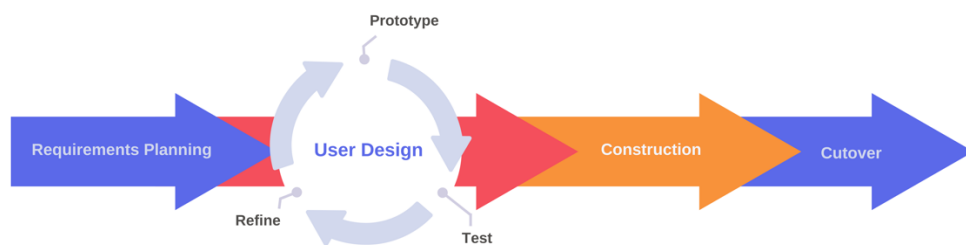
#### **2.4. *Rapid Application Development* (RAD)**

*Rapid Application Development* (RAD) adalah pendekatan pengembangan perangkat lunak yang menekankan kecepatan dan fleksibilitas. Metode ini relatif mudah diterapkan karena setiap tahap pengembangan berfokus pada persyaratan spesifik, sehingga menghasilkan waktu pengembangan yang lebih singkat. Selain itu, tahap-tahap dalam RAD dirancang dengan siklus pengembangan yang lebih singkat, memungkinkan hasil perangkat lunak dapat dilihat secara langsung tanpa harus menunggu lama (Suryasari et al., 2022).

Keuntungan utama metode RAD terletak pada kesesuaiannya untuk proyek dengan batas waktu yang ketat. Meskipun masih mengikuti tahapan umum pengembangan sistem, RAD memiliki keunggulan dalam memanfaatkan komponen perangkat lunak yang sudah ada. Dengan demikian, pengembang tidak

perlu memulai proses dari awal, menghemat waktu, tenaga, dan biaya, sekaligus menghasilkan proses pengembangan yang lebih efisien (Fauzi & Iksari, 2023).

Selain itu, RAD menekankan pentingnya kolaborasi antara pengembang dan pengguna akhir. Pengguna secara aktif terlibat dalam setiap tahap pengembangan untuk memberikan umpan balik konstruktif, memastikan bahwa sistem yang dibangun benar-benar memenuhi kebutuhan mereka. Keterlibatan langsung pengguna ini tidak hanya meningkatkan kualitas sistem tetapi juga mengurangi risiko ketidakpuasan setelah implementasi sistem. (Schuurman et al., 2022).



**Gambar 2.2. Model RAD**

*Sumber :* (Suryasari et al., 2022).

Tahap pertama dalam proses RAD adalah *Requirement Planning* (Perencanaan Kebutuhan). Pada tahap ini, tim pengembang bersama dengan pengguna berkolaborasi untuk merumuskan tujuan proyek, mengidentifikasi fitur serta fungsi yang diperlukan, dan menetapkan sumber daya yang dibutuhkan dalam proses pengembangan. Perencanaan kebutuhan yang matang berperan penting dalam memastikan proyek memiliki arah yang jelas, terukur, serta tujuan yang realistis untuk dicapai (Suryasari et al., 2022).

Tahap berikutnya adalah desain sistem. Dalam RAD, desain dilakukan secara iteratif, dengan prototipe yang dikembangkan dan dievaluasi oleh pengguna. Setiap iterasi dalam RAD melibatkan pembuatan prototipe baru atau penyempurnaan prototipe yang sudah ada berdasarkan umpan balik pengguna. Pendekatan iteratif ini memberikan fleksibilitas dalam menanggapi perubahan persyaratan dan memastikan sistem berkembang sesuai dengan harapan pengguna (Beatriks et al., 2025) .

Tahap *Construction*, fokus utama adalah pengembangan sistem. Dalam metode RAD, pengguna tetap terlibat secara aktif dan dapat memberikan saran untuk perubahan atau perbaikan sepanjang proses. Pengembang bertanggung

jawab atas pengembangan sistem, pemrograman, dan pengujian. Proses pengembangan dilakukan dalam siklus singkat yang disebut iterasi, di mana setiap iterasi menghasilkan versi sistem yang dapat diuji. Pendekatan iteratif ini memungkinkan tim pengembangan untuk mengidentifikasi dan memperbaiki masalah dengan lebih cepat, sambil mengakomodasi perubahan persyaratan dengan lebih efektif (Sirait et al., 2023).

Terakhir *Cutover*, pada tahap terakhir dalam implementasi RAD, sistem fungsional diuji menggunakan metode *User Acceptance Test (UAT)* dengan model pengujian black-box testing dan akan ditransfer ke sistem yang baru dan dilanjutkan dengan pelatihan pengguna. Selain itu, akan dilakukan perbandingan antara RAD dan metode tradisional, dan hasilnya, sistem baru harus dapat diimplementasikan di perusahaan (Editors & Editors, n.d.).

Pemilihan kerangka kerja yang tepat dalam pengembangan sistem akademik juga sangat penting untuk memastikan kesesuaian dengan kebutuhan, skalabilitas, dan ketersediaan sumber daya.

Adapun kelebihan dan kekurangan RAD pada tabel dibawah:

Kelebihan	Kekurangan
Metode ini berfokus pada siklus pengembangan yang dirancang agar berlangsung lebih cepat, namun tetap mampu menghasilkan kualitas yang lebih baik dibandingkan dengan metode pengembangan perangkat lunak lainnya.	Metode ini cenderung kurang memberikan perhatian pada tahap perencanaan dan lebih berfokus pada proses pengembangan di tahap-tahap berikutnya.
Metode ini sangat cocok digunakan dalam pengembangan perangkat lunak yang memiliki permintaan mendesak dari pengguna dan harus diselesaikan dalam waktu yang relatif singkat.	Dalam proses pengembangan perangkat lunak atau aplikasi, tidak semua jenis proyek cocok untuk menggunakan metode RAD. Hal ini disebabkan oleh potensi masalah yang timbul jika modul-modul dalam sistem atau aplikasi tidak terorganisir dengan baik, sehingga penerapan model RAD dapat menimbulkan hambatan.
Model RAD memiliki kemampuan adaptasi yang baik terhadap sistem yang telah ada sebelumnya, sehingga memudahkan proses penyesuaian komponen-	Metode RAD kurang tepat diterapkan pada pengembangan sistem yang memiliki tingkat risiko sangat tinggi, karena memerlukan pengendalian dan pengawasan

komponennya agar sesuai dengan kebutuhan pengembangan.	yang lebih ketat di setiap tahap pengembangannya.
--	---

**Tabel 2.1. Kelebihan dan Kekurangan RAD**

Sumber((Informasi et al., 2023))(Pricillia & Zulfachmi, 2021)

## 2.5. Fase-Fase Metode *Rapid Application Development (RAD)*

Metode *Rapid Application Development (RAD)* diterapkan dalam proses pengembangan aplikasi atau sistem dengan menekankan pada beberapa fase utama yang berlangsung secara cepat dan terstruktur. Menurut Atina (2021), terdapat tiga fase utama dalam metode RAD, yaitu:

### 1. *Requirement Planning*:

Pada tahap ini, tim pengembangan fokus pada pemahaman persyaratan sistem dengan mengidentifikasi kebutuhan informasi dan masalah yang dihadapi. Selain itu, tujuan, batasan, kendala, dan solusi alternatif yang mungkin juga ditentukan. Proses analisis dilakukan untuk memahami perilaku sistem dan berbagai aktivitas yang terjadi di dalamnya, sehingga persyaratan pengguna dapat diformulasikan dengan jelas dan terstruktur.

### 2. *Design Workshop*:

Tahap ini melibatkan identifikasi berbagai solusi alternatif dan pemilihan solusi yang paling sesuai untuk implementasi. Selanjutnya, proses bisnis dirancang dan diprogram berdasarkan data yang dimodelkan dalam arsitektur sistem. Pada tahap ini, salah satu alat yang sering digunakan untuk pemodelan sistem adalah *Unified Modeling Language (UML)*, yang memudahkan representasi visual yang jelas tentang struktur dan alur sistem.

### 3. *Implementation*:

Setelah fase desain selesai, langkah berikutnya adalah mengimplementasikan sistem yang telah dirancang. Proses ini melibatkan penulisan kode program dalam format yang dapat dieksekusi oleh komputer untuk menciptakan program atau unit sistem yang berfungsi sepenuhnya. Tujuan utama fase ini adalah memastikan bahwa sistem siap digunakan dan dapat beroperasi secara optimal sesuai dengan persyaratan pengguna.

## 2.6. Kerangka Kerja

Selama dua puluh tahun terakhir, banyak sekali kerangka kerja yang telah membuahkan hasil dalam industri pengembangan perangkat lunak. Setiap kerangka kerja telah hadir dengan keistimewaannya sendiri, ceruknya yang membuatnya sedikit lebih hebat dari pendahulunya. Memilih salah satu berarti memilih gaya pengembangan yang disukai seperti bahasa, infrastruktur, arsitektur, basis data integrasi, alat pihak ketiga, fitur-fitur khusus. Semua ini ditentukan oleh kerangka kerja (Richard, 2022).

Penting untuk dipahami bahwa tidak ada kerangka kerja yang lebih baik dari yang lain melainkan, dalam basis bahasa mereka, kompatibilitas dengan perpustakaan lain, kemudahan penggunaan, popularitas, dan dokumentasi, masing-masing kerangka kerja benar-benar menemukan keunikannya masing-masing.

Perkembangan komputer dan internet saat ini juga didukung oleh berbagai aplikasi berbasis web yang menggunakan kerangka kerja (*framework*). Kerangka kerja adalah kumpulan aturan untuk pengembangan perangkat lunak, yang dirancang untuk mengurangi kesalahan dalam aktivitas yang sering dilakukan selama proses pengembangan situs web (Sotnik et al., 2023).

Saat ini, terdapat banyak kerangka kerja pemrograman yang tersedia dan dapat digunakan untuk membangun program atau aplikasi. Kerangka kerja dikembangkan untuk menyediakan fungsionalitas spesifik yang memudahkan programmer dalam membuat program dengan lebih cepat dan efisien. Hampir setiap bahasa pemrograman modern memiliki kerangka kerjanya sendiri. Penggunaan kerangka kerja tidak hanya mempercepat waktu pengembangan, tetapi juga memudahkan kolaborasi antara programmer, karena kerangka kerja menyediakan pedoman dan standar untuk penulisan kode program. Beberapa contoh kerangka kerja yang sering digunakan antara lain CodeIgniter, CakePHP, Symfony, Kohana, dan Zend Framework (Prasena et al., 2020) .

Salah satu framework yang banyak direkomendasikan di komunitas pemrograman adalah Laravel (PRASANDY et al., 2021). Situs web yang dibangun menggunakan Laravel memiliki lebih banyak baris kode yang berorientasi pada fungsi dibandingkan dengan PHP Native, sehingga menghasilkan efisiensi yang lebih tinggi dalam pembuatan fungsi kode. Hal ini karena Laravel menyediakan berbagai perpustakaan, termasuk penggunaan ORM (Object Relational Mapping) untuk manajemen database, sehingga mengurangi waktu pengembangan dan membuat sistem lebih mudah untuk dipelihara (Gufon et al., 2021).

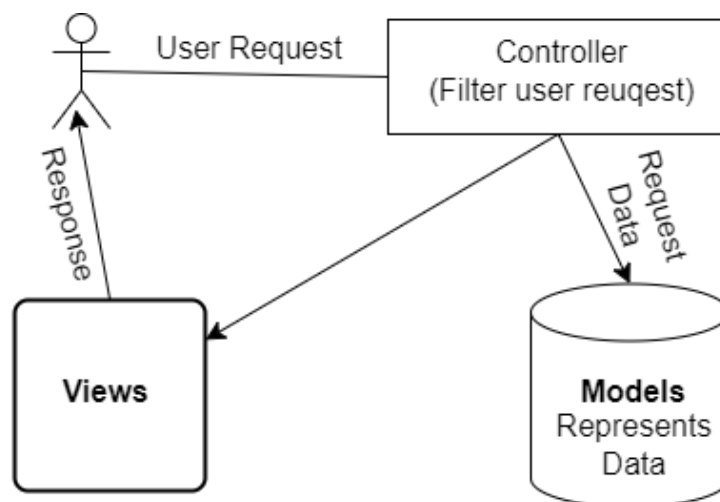
Menurut Google Trends, sepanjang tahun 2022 Laravel menempati peringkat lebih tinggi daripada kerangka kerja lainnya (Kansha, 2023). Selain itu, JetBrains melaporkan bahwa Laravel adalah kerangka kerja PHP yang paling banyak digunakan dengan pangsa pasar 67% pada tahun 2021. Hal ini menunjukkan bahwa Laravel, meskipun merupakan framework PHP yang relatif baru, berhasil menduduki peringkat lebih tinggi daripada pendahulunya CodeIgniter dan tetap menjadi tren

utama dalam beberapa tahun terakhir (Prasena et al., 2020). Berdasarkan hal ini, proses pengembangan sistem informasi akademik di YP Ummi Lubuk Pakam menggunakan framework Laravel.

## 2.7. Laravel

Laravel adalah salah satu framework PHP paling populer dan banyak digunakan dalam pengembangan aplikasi web. Framework ini dikembangkan oleh Taylor Otwell dan pertama kali diperkenalkan pada tahun 2011. Laravel mengimplementasikan arsitektur *Model-View-Controller* (MVC), sebuah pola desain yang membagi aplikasi menjadi tiga komponen utama: Model, View, dan Controller. Komponen View berperan dalam menampilkan antarmuka kepada pengguna, sedangkan *Controller* bertugas menerima serta memproses data masukan, sekaligus menjadi penghubung antara Model dan View. Sementara itu, Model menangani logika dan pengelolaan data yang berkaitan dengan pengguna, serta berfungsi sebagai representasi utama data yang ditransfer antara *View* dan *Controller* (Subecz, 2021).

Salah satu faktor yang membuat Laravel banyak diminati oleh para pengembang adalah tingkat kemudahan penggunaannya serta dokumentasinya yang lengkap. Laravel menyediakan berbagai fitur dan alat yang mempermudah proses pengembangan aplikasi web, termasuk konfigurasi routing, sistem autentikasi, dan pengelolaan database (Petrovic et al., 2020). Ketersediaan fitur-fitur tersebut memungkinkan pengembang untuk menghemat waktu, mempercepat proses pembuatan aplikasi, dan meningkatkan produktivitas secara keseluruhan.



### **Gambar 2.3. MVC Architecture**

Sumber: (Subecz, 2021).

*Routing* adalah salah satu fitur penting dalam Laravel yang memungkinkan pengembang untuk mendefinisikan URL dan tindakan yang akan dilakukan ketika URL tersebut diakses. Dengan menggunakan *routing*, pengembang dapat mengatur alur kerja aplikasi dengan mudah dan membuat aplikasi yang responsif dan dinamis. Laravel juga mendukung *routing* berbasis RESTful, yang memungkinkan pengembangan aplikasi yang sesuai dengan prinsip-prinsip REST (Petrovic et al., 2020).

*Eloquent* merupakan fitur *Object-Relational Mapping* (ORM) yang terintegrasi dalam framework Laravel, yang memungkinkan pengembang berinteraksi dengan basis data melalui model berbasis objek. Dengan menggunakan *Eloquent*, pengembang dapat melakukan berbagai operasi CRUD (Create, Read, Update, Delete) pada tabel basis data dengan lebih mudah, tanpa perlu menulis perintah SQL secara manual. Selain itu, *Eloquent* mendukung berbagai jenis hubungan antara tabel, seperti satu-ke-satu, satu-ke-banyak, dan banyak-ke-banyak, sehingga memudahkan pengelolaan hubungan antara entitas dalam basis data (Hariyanto et al., 2021).

Laravel juga memiliki ekosistem yang luas dengan berbagai paket dan alat tambahan yang dapat dimanfaatkan untuk memperluas fungsionalitas aplikasi. Salah satu paket yang cukup populer adalah Laravel Nova, yaitu alat administrasi yang membantu pengembang dalam membangun antarmuka pengguna yang intuitif untuk mengelola basis data. Laravel Nova menyediakan berbagai komponen seperti tabel, formulir, dan grafik, yang mempermudah serta mempercepat proses pengembangan antarmuka administrasi (Suryasari et al., 2022).

Selain itu, Laravel juga mendukung pengujian otomatis (*automated testing*) yang memungkinkan pengembang menulis dan menjalankan berbagai tes untuk memastikan aplikasi berfungsi sesuai dengan spesifikasi. *Framework* ini menyediakan beragam alat dan pustaka untuk melakukan unit testing, integration testing, serta functional testing. Dengan penerapan pengujian otomatis, pengembang dapat mendeteksi dan memperbaiki kesalahan (*bug*) lebih cepat, serta memastikan bahwa setiap pembaruan tidak mengganggu fungsionalitas yang telah ada (Li, 2020).

## **2.8. Unified Modeling Language (UML)**

Beberapa sudut pandang arsitektur perangkat lunak memodulasi arsitektur perangkat lunak dalam berbagai sudut pandang yang berbeda yang masing-masing menangani masalah yang berbeda. Unified Modeling Language (UML)



sangat populer di kalangan praktisi untuk memodelkan arsitektur perangkat lunak dari berbagai sudut pandang (Ozkaya & Erata, 2020).

*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, mendefinisikan, dan mengembangkan perangkat lunak. UML berfungsi sebagai metodologi dalam pengembangan sistem berorientasi objek serta sebagai alat dalam proses tersebut. Bahasa ini menggunakan representasi grafis untuk memvisualisasikan, menjelaskan, mengembangkan, dan mendokumentasikan sistem perangkat lunak *Object-oriented* (OO) (Voutama, 2022).



Selain itu, UML menyediakan standar penulisan yang memungkinkan pembuatan blueprint sistem yang komprehensif, mencakup konsep proses bisnis, struktur kelas dalam bahasa pemrograman tertentu, desain skema basis data, dan komponen lain yang diperlukan dalam pengembangan perangkat lunak. Beberapa jenis diagram yang terdapat dalam UML meliputi: merupakan bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menspesifikasikan, dan membangun perangkat lunak (Suendri, 2018):

### 2.8.1. Use Case Diagram

Diagram *Use Case* adalah teknik pemodelan yang digunakan untuk menggambarkan perilaku sistem yang akan dikembangkan. Diagram ini menggambarkan interaksi antara satu atau lebih aktor dan sistem yang bersangkutan (Language, 2024).

Simbol	Deskripsi
	Abstraksi dan interaksi antara sistem dan aktor
	Mewakili peran yang dijalankan oleh manusia, sistem lain, atau perangkat ketika berinteraksi dengan use case.
Asosiasi/ <i>association</i>	Abstraksi dari penghubung antara aktor dengan <i>use case</i>


**Tabel 2.2. Simbol-simbol Use Case Diagram**

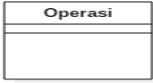



<p>Extend</p> 	<p>Menunjukkan bahwa suatu use case dapat berfungsi sebagai tambahan fungsional dari use case lainnya apabila kondisi tertentu terpenuhi.</p>
<p><i>Include</i></p>	<p>Spesialisasi aktor</p>
	<p>Menunjukkan adanya spesialisasi pada aktor sehingga dapat berpartisipasi dalam use case tertentu.</p>

Sumber: Dicoding Intern 2021

### 2.8.2. Class Diagram

*Class diagram* adalah representasi visual yang menggambarkan struktur suatu sistem dengan mendefinisikan kelas-kelas yang digunakan dalam proses pengembangan sistem tersebut. Diagram ini menampilkan jenis-jenis objek dalam sistem dan hubungan statis atau interaksi antara objek-objek tersebut.

Simbol	Deskripsi
	<p>Merupakan suatu wadah yang mengelompokkan satu atau lebih kelas dalam struktur sistem.</p>






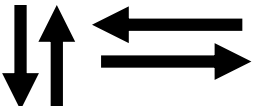
	Kelas struktur sistem
<p>Antar muka / <i>interface</i></p> 	Memiliki konsep yang setara dengan <i>interface</i> dalam pemrograman berorientasi objek.
<p>Asosiasi</p> 	Relasi umum pada kelas
<p>Asosiasi berarah/<i>directed association</i></p> 	Menunjukkan bahwa satu kelas digunakan oleh kelas lain, di mana asosiasi ini umumnya juga disertai dengan multiplicity.

**Tabel 2.3. Simbol –Simbol Class Diagram**

*Sumber : Ismail, 2019*

### 2.8.3. Activity Diagram

*Activity diagram* adalah diagram aktivitas adalah jenis diagram yang berfokus pada aliran aktivitas atau proses di dalam suatu sistem. Diagram ini menampilkan urutan tindakan yang dipicu oleh kondisi sebelumnya (pemrosesan internal). Oleh karena itu, diagram aktivitas tidak digunakan untuk menggambarkan perilaku internal suatu sistem atau interaksi antara subsistem secara rinci, melainkan untuk memvisualisasikan aliran proses dan jalannya aktivitas dari satu tahap ke tahap berikutnya pada tingkat tertentu.

Tabel	Nama	Keterangan
	<i>Activity</i>	Menunjukkan bagaimana setiap kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	Mencerminkan eksekusi suatu aksi dalam sistem.
	<i>Initial Node</i>	Pembentukan objek
	<i>Activity final Node</i>	Penghentian objek
	<i>Decision</i>	Digunakan untuk menggambarkan keputusan atau tindakan yang harus diambil berdasarkan kondisi tertentu.
	<i>Line Connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya dalam diagram.

**Tabel 2.4. Simbol-simbol *Activity* Diagram**

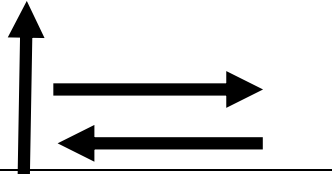
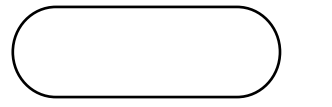
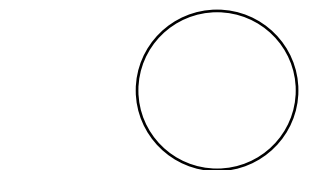
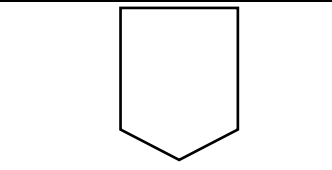

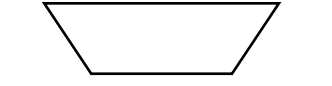
*Sumber : Dicoding Intern, 2021*

#### 2.8.4. Flowchart

Flowchart merupakan representasi grafis yang menggambarkan langkah-langkah dan urutan prosedur dalam suatu program atau sistem. Diagram alir ini berfungsi sebagai teknik analisis visual yang digunakan untuk menjelaskan berbagai aspek sistem informasi secara jelas, ringkas, dan logis. Melalui diagram

alir, dapat ditunjukkan bagaimana proses bisnis berlangsung dan bagaimana dokumen dan data mengalir di dalam organisasi. Selain itu, diagram alir juga mencerminkan sistem, prosedur, dan pengendalian internal yang diterapkan oleh perusahaan (Akuntansi et al., 2023).

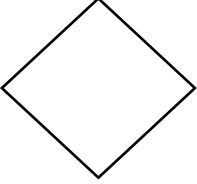
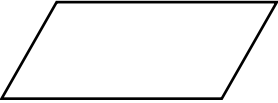

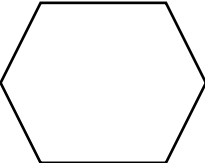

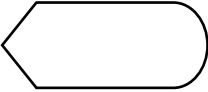
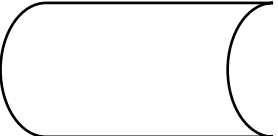
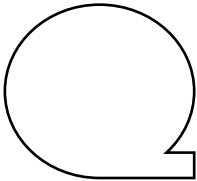
Diagram alir sistem menggambarkan urutan proses dalam suatu sistem dengan menampilkan media masukan, keluaran, dan jenis media penyimpanan yang digunakan dalam pemrosesan data. Sementara itu, diagram alir program menampilkan urutan proses secara lebih rinci dengan menggunakan simbol-simbol khusus untuk menunjukkan hubungan antara instruksi dalam program. Dengan demikian, diagram alir tidak hanya memudahkan pemahaman logika sistem tetapi juga mendukung pengambilan keputusan yang lebih akurat. (Fingerhut et al., 2021).

	<i>Flow Direction Symbol</i>	Simbol yang digunakan untuk menghubungkan satu simbol dengan simbol lainnya.
	<i>Terminator Symbol</i>	Simbol yang menandai awal ( <i>start</i> ) atau akhir ( <i>stop</i> ) dari suatu kegiatan.
	Connector Symbol	Simbol untuk keluar-masuk atau penyambungan proses pada lembar/halaman yang sama.
	Connector Symbol	Simbol untuk keluar-masuk atau penyambungan proses pada lembar/halaman yang berbeda.
	Processing Symbol	Simbol yang menunjukkan pengolahan yang dilakukan oleh computer
	Simbol Manual Operation	Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer



**Tabel 2.5 Simbol**

**Flowchart**

Sumber

	Simbol Decision	Simbol pemilihn proses berdasarkan kondisi yang ada
	Simbol Input-Output	Simbol yang menyatakan roses input dan output tanpa tergantung dengan jenis peralatannya
	Simbol Manual Input	Simbol untuk pemasukan data secara manual on-line keyboard
	Simbol Preparation	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage
	Simbol Predefine Proses	Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Simbol Display	Simbol untuk pelaksanaan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Simbol Disk and On-Line Storage	Simbol yang menyatakan input yang berasal dari isk atau disimpan ke disk
	Simbol Magnetik Tape Unit	Simbol yang menyatakan input berasal dari pita magnetic atau output disimpan ke pita magnetic

(Tuasamu, et al., 2023)

	Simbol Punch Card	Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Dokumen	Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas

## 2.9. Xampp

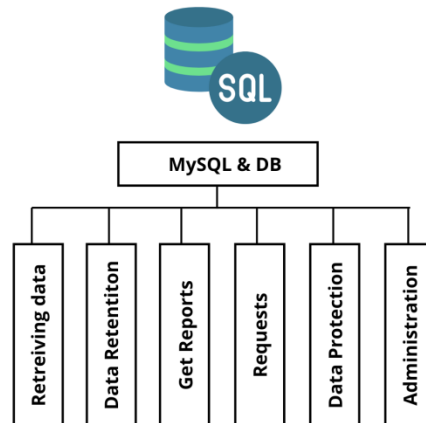
XAMPP merupakan singkatan dari X (Cross-Platform), Apache, MySQL, PHP, dan Perl, yang merupakan paket perangkat lunak yang mengintegrasikan berbagai komponen penting untuk pengembangan aplikasi web. Dengan menginstal XAMPP, pengguna tidak perlu lagi menginstal dan mengonfigurasi server web Apache, PHP, atau MySQL secara manual, karena semua komponen ini disediakan dalam paket terintegrasi. Selain itu, XAMPP juga dilengkapi dengan berbagai komponen pendukung, seperti server FTP, PhpMyAdmin, dan perpustakaan tambahan lainnya, yang mempermudah proses pengembangan aplikasi web.

XAMPP dikenal sebagai aplikasi server web instan dan komprehensif karena menyediakan semua yang diperlukan untuk menjalankan situs web berbasis Sistem Manajemen Konten (CMS) seperti Joomla atau WordPress. Sebagai paket penginstal AMP (Apache, MySQL, PHP), XAMPP sangat mudah digunakan, terutama bagi pengguna yang belum memiliki server sendiri. Melalui XAMPP, pengguna dapat menjalankan, menguji, dan mengembangkan situs web berbasis server dan database secara lokal sebelum mengunggahnya ke server sebenarnya (Nirsal et al., 2020).

## 2.10. MySQL

MySQL adalah sistem manajemen basis data relasional, yang berarti data disimpan dalam beberapa tabel terpisah namun saling terhubung, memungkinkan manipulasi dan pengambilan data yang lebih cepat dan efisien. MySQL berfungsi sebagai program manajemen basis data yang digunakan untuk memproses, menyimpan, dan memanipulasi data di server. Nama MySQL berasal dari *Structured Query Language* (SQL), bahasa terstruktur yang dirancang khusus untuk manajemen basis data. Dengan arsitektur relasionalnya, MySQL memfasilitasi pengelolaan data yang terorganisir dan efisien dalam lingkungan server.

Dalam berbagai penelitian, MySQL dikenal sebagai salah satu sistem database yang paling populer di kalangan pengembang web. Kepopuleran ini disebabkan oleh kemampuannya dalam melakukan pencarian, penambahan, penghapusan, dan pengurutan data dengan cepat serta mudah. Selain itu, MySQL juga menawarkan sistem keamanan yang sederhana namun andal, serta tersedia secara gratis di bawah lisensi terbuka, menjadikannya pilihan utama bagi banyak pengembang (Sotnik et al., 2023).



**Gambar 2.4. Basic MySQL Functions**

Sumber: (Sotnik et al., 2023)

MySQL Database Server merupakan sistem basis data lintas platform yang bersifat *open source*. Banyak situs web berskala besar memanfaatkan MySQL sebagai tempat penyimpanan data utama karena kemampuannya yang efisien, stabil, dan handal dalam mengelola volume data yang besar secara efektif.

## 2.11. API

*Application Programming Interface* (API) merupakan sekumpulan aturan dan mekanisme yang memungkinkan dua sistem perangkat lunak berbeda saling berinteraksi dan bertukar informasi. API berperan sebagai jembatan penghubung antara aplikasi client dan server, di mana client dapat meminta layanan atau data tanpa perlu memahami detail teknis dari sistem server tersebut (Perkasa & Setiawan, 2018).

Dalam pengembangan aplikasi berbasis web maupun mobile, jenis API yang paling sering digunakan adalah REST (Representational State Transfer) API. REST menggunakan protokol HTTP serta memanfaatkan metode standar seperti GET, POST, PUT, dan DELETE untuk melakukan berbagai operasi terhadap

sumber daya (*resource*). Keunggulan dari REST terletak pada sifatnya yang stateless (tidak menyimpan status sesi) serta fleksibilitasnya dalam mengirim dan menerima data dalam format ringan seperti JSON, yang mudah dibaca dan diolah oleh sistem (Utama & Sari, 2022).

Dalam sistem pengingat pembayaran SPP berbasis WhatsApp, API memiliki dua peran utama:

- API Internal (Server Side): API yang dikembangkan sendiri untuk sistem sekolah. API ini bertanggung jawab untuk mengambil data tagihan (Nomor Tagihan, Nama Siswa, Jumlah, Tanggal Jatuh Tempo) dari Database (MySQL) dan memprosesnya.
- API Eksternal (WhatsApp Business API): Ini adalah API yang disediakan oleh penyedia layanan (Meta/Partner WhatsApp). Sistem internal akan mengirimkan permintaan (POST Request) ke WhatsApp API dengan menyertakan data siswa dan konten pesan template yang sudah disiapkan. WhatsApp API kemudian bertugas mengirimkan pesan tersebut ke nomor tujuan.

Peran API ini memastikan bahwa proses pengiriman notifikasi berlangsung secara otomatis, terstruktur, dan real-time setelah data tagihan diperbarui di database (Saputra & Safitri, 2022).

## 2.12. **Black-Box Testing**

Dalam pengujian aplikasi yang dikembangkan, metode *Black-Box Testing* digunakan untuk memastikan bahwa sistem beroperasi sesuai dengan spesifikasi yang ditetapkan. Metode ini tidak memerlukan akses ke kode sumber, sehingga pengujian berfokus pada masukan yang diberikan dan keluaran yang dihasilkan oleh sistem, sesuai dengan kebutuhan pengguna akhir. *Black-Box Testing* dikenal sebagai pengujian berbasis perilaku (*behavioral testing*) karena penguji hanya menilai fungsionalitas aplikasi dari luar, tanpa mengetahui struktur internal perangkat lunak (Putri & Sari, 2021).

Metode ini sangat cocok diterapkan dalam pengujian aplikasi yang telah selesai dikembangkan, karena dapat mensimulasikan interaksi pengguna secara langsung. Beberapa teknik yang umum digunakan dalam pendekatan ini antara lain analisis partisi ekuivalensi, analisis nilai batas (*boundary value analysis*), pengujian transisi status (*state transition testing*), dan *decision table testing* (Wahyuni & Nugroho, 2022).

Penggunaan *Black-Box Testing* membantu menemukan kekurangan dalam pemenuhan kebutuhan pengguna, yang sering kali tidak dapat diidentifikasi hanya melalui pengujian berbasis kode. Namun, metode ini juga memiliki keterbatasan, seperti kurangnya informasi mendalam tentang logika internal program yang dapat

menyebabkan beberapa skenario kegagalan terlewat (Anjani & Ramadhan, 2020). Meskipun demikian, pendekatan ini tetap menjadi pilihan yang tepat dalam tahap akhir pengembangan, khususnya dalam mengevaluasi aplikasi dari sudut pandang pengguna.

### 2.13. *State of The Art*

Pada tabel *State of The Art* di bawah ini disajikan kumpulan jurnal yang diperoleh melalui Google Scholar, yang memuat hasil-hasil penelitian dari berbagai peneliti sebelumnya. Jurnal-jurnal tersebut memiliki keterkaitan dengan penelitian yang sedang dilakukan, sehingga dapat menjadi dasar perbandingan, referensi, dan penguat dalam pengembangan penelitian ini.

No	Judul Jurnal	Nama Penulis dan Tahun Terbit	Hasil	Metodologi
1	Penerapan Metode RAD pada Sistem Akademik di Sekolah Menengah	Setiawan & Rahmawati (2023)	Sistem lebih responsif, waktu pengembangan lebih pendek dibanding metode konvensional.	Kualitatif
2	<i>Pengembangan Sistem Informasi Akademik Menggunakan Metode RAD</i>	Dewi & Hartono (2020)	Sistem dapat dikembangkan dalam waktu lebih singkat dengan fitur yang sesuai kebutuhan pengguna.	Kualitatif
3	<i>Pengembangan Sistem Akademik Berbasis Website Menggunakan RAD</i>	Hasanah & Fadillah (2021)	Penggunaan RAD meningkatkan kepuasan pengguna dan efektivitas input data siswa.	Kualitatif
4	Implementasi RAD dalam Sistem Akademik Berbasis Web	Kurniawan & Putri (2021)	RAD mempercepat proses pengembangan sistem akademik dan meningkatkan efisiensi pengelolaan data akademik.	Kualitatif
5	Rancang Bangun Sistem Akademik Berbasis Web dengan RAD	Wulandari et al. (2022)	RAD mampu menyesuaikan kebutuhan pengguna yang berubah secara cepat dan iteratif.	Kualitatif

**Tabel 2.6. *State of The Art***