

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian pertama yang dilakukan oleh (Sangga Rasefta & Esabella, 2020) mengembangkan Sistem Informasi Akademik SMK Negeri 3 Sumbawa Besar berbasis Web dengan metode *Extreme Programming* (XP). Sistem ini dibangun menggunakan PHP dan MySQL serta diuji dengan *Black-box Testing*, memastikan peningkatan efisiensi pengelolaan akademik. Tujuan utama penelitian ini adalah membantu bagian kurikulum dalam mengelola informasi akademik sekolah secara lebih efektif dan efisien.

Penelitian kedua yang dilakukan oleh (Ratulangi dkk., 2021) mengembangkan Sistem Informasi Akademik SMA Negeri 3 Tondano berbasis Web dengan metode Prototipe. Sistem ini membantu pengolahan data akademik secara lebih efektif dan efisien serta diuji dengan *Black-box Testing* yang menunjukkan validitas fungsi sistem. Sistem ini mendukung pengelolaan data akademik, termasuk data guru, siswa, kelas, mata pelajaran, jadwal, dan nilai.

Penelitian ketiga yang dilakukan oleh (Lukman Santoso & Juni Amanullah, 2022) mengembangkan Sistem Informasi Akademik SMA YATPI Godong berbasis Web menggunakan metode *Rapid Application Development* (RAD). Sistem ini mendukung transparansi data akademik, absensi, dan pembayaran SPP, serta diuji dengan *Black Box Testing* yang menunjukkan keberhasilan implementasi. Pengembangan sistem dilakukan dengan PHP dan MySQL, serta bertujuan untuk meningkatkan efisiensi pengelolaan data akademik di SMA YATPI Godong.

Penelitian keempat yang dilakukan oleh (Gunawan & Kraugusteeliana, 2023) merancang Sistem Informasi Akademik SMP Negeri 1 Warungkiara berbasis Web dengan metode *Scrum*. Penelitian ini bertujuan untuk meningkatkan efisiensi pengelolaan data akademik, terutama dalam hal pemberian informasi kepada siswa dan pencatatan nilai oleh guru. Sistem ini diuji dan dinyatakan layak digunakan karena dapat mengelola data akademik dengan lebih baik serta membantu guru dalam mengolah nilai siswa.

Penelitian kelima yang dilakukan oleh (Luh Gede Pivin Suwirmayanti dkk., 2023) meneliti implementasi *framework* Laravel pada Sistem Informasi Akademik SMA Negeri 1 Kediri berbasis web dengan metode *Waterfall*. Penelitian ini menghasilkan sistem yang

memungkinkan admin mengelola data siswa, guru, kelas, dan jadwal. Guru dapat mengolah biodata, nilai, serta melihat daftar siswa di kelasnya. Siswa hanya memiliki akses untuk melihat jadwal dan nilai akademik. Perancangan sistem menggunakan *Data Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD), dan desain antarmuka.

Tabel 2.1 Penelitian Terdahulu

Peneliti (Tahun)	Judul Penelitian	Metode	Hasil Penelitian
(Sangga Rasefta & Esabella, 2020)	Sistem Informasi Akademik SMK Negeri 3 Sumbawa Besar Berbasis Web	<i>Extreme Programmin</i> g (XP)	Sistem berbasis PHP dan MySQL telah diuji dengan <i>Black Box Testing</i> dan diterapkan untuk meningkatkan mutu pendidikan.
(Ratulangi dkk., 2021)	Sistem Informasi Akademik Berbasis Web di SMA Negeri 3 Tondano	Prototipe	Sistem diuji menggunakan <i>Black Box Testing</i> dan membantu pengolahan data akademik menjadi lebih efektif dan efisien.
(Lukman Santoso & Juni Amanullah, 2022)	Pengembangan Sistem Informasi Akademik Berbasis Web Menggunakan Metode <i>Rapid Application Development</i> (RAD)	<i>Rapid Application Development</i> (RAD)	Sistem diuji dengan metode <i>Black Box Testing</i> dan telah diterapkan di SMA YATPI Godong untuk meningkatkan kualitas akademik.

Peneliti (Tahun)	Judul Penelitian	Metode	Hasil Penelitian
(Gunawan & Kraugusteeliana, 2023)	Rancang Bangun Sistem Informasi Akademik Sekolah Berbasis Web di SMP Negeri 1 Warungkiara	<i>Scrum</i>	Sistem berhasil digunakan untuk meningkatkan pengelolaan data akademik serta membantu guru dalam pengolahan nilai siswa.
(Luh Gede Pivin Suwirmayanti dkk., 2023)	Implementasi <i>framework</i> Laravel Pada Sistem Informasi Akademik SMA Negeri 1 Kediri Berbasis Web	<i>Waterfall</i>	Sistem terdiri dari tiga level pengguna (admin, guru, siswa) dan menggunakan DFD, ERD, serta desain antarmuka untuk perancangannya.

Berbagai penelitian sebelumnya telah menghasilkan sistem informasi akademik berbasis web yang efektif dan efisien dengan menggunakan metode pengembangan perangkat lunak seperti *Waterfall*, *Rapid Application Development* (RAD), *Scrum*, *Prototipe*, dan *Extreme Programming*. Namun, penelitian-penelitian tersebut belum banyak yang menerapkan metode *Prototyping*, yang menawarkan pendekatan iteratif dengan keterlibatan aktif pengguna dalam setiap tahap pengembangan (Purnomo, 2017). Dengan adanya penelitian ini, diharapkan sistem informasi akademik berbasis web yang dikembangkan menggunakan metode *Prototyping* dapat memberikan kontribusi signifikan dalam pengelolaan data akademik yang lebih responsif terhadap kebutuhan pengguna dan fleksibel terhadap perubahan.

2.2 Landasan Teori

2.2.1 Sistem Informasi Berbasis Web

Sistem informasi berbasis web adalah sistem yang memanfaatkan teknologi web untuk mengelola, menyimpan, dan mendistribusikan informasi. Menurut (Alamsyah dkk., 2021), sistem ini memungkinkan akses informasi secara luas tanpa batasan geografis, sehingga memudahkan pengguna dalam mendapatkan data yang dibutuhkan secara *real-time*. Penggunaan sistem informasi berbasis web sangat penting dalam berbagai bidang, termasuk pendidikan, bisnis, dan pemerintahan. Sistem informasi berbasis web terdiri dari beberapa komponen utama, yaitu:

- 1) *Frontend*: Antarmuka pengguna yang berfungsi sebagai titik interaksi antara pengguna dan sistem.
- 2) *Backend*: Server yang mengelola logika bisnis dan penyimpanan data.
- 3) *Database*: Tempat penyimpanan data yang digunakan oleh sistem.
- 4) Jaringan: Infrastruktur yang memungkinkan komunikasi antara *frontend* dan *backend*.

(Naibaho, 2024) menekankan bahwa desain sistem yang baik akan memperhatikan semua komponen ini agar dapat berfungsi secara optimal.

Metode pengembangan sistem informasi berbasis web sering kali menggunakan model pengembangan seperti *Waterfall* atau *V-Model*. *Model Waterfall* adalah pendekatan berurutan yang mencakup analisis kebutuhan, desain, pengkodean, dan pengujian (Sutejo & Alo, 2024). (Sutejo & Alo, 2024) menjelaskan bahwa *Model* ini cocok untuk proyek dengan persyaratan yang jelas dan tidak berubah selama proses pengembangan.

Sistem informasi berbasis web menawarkan berbagai keuntungan, antara lain:

- 1) Aksesibilitas: Dapat diakses dari mana saja dengan koneksi internet.
- 2) Biaya Efisien: Mengurangi biaya infrastruktur karena tidak memerlukan perangkat keras khusus.
- 3) Kemudahan Pemeliharaan: Pembaruan sistem dapat dilakukan secara terpusat tanpa perlu menginstal ulang di setiap perangkat pengguna.

Meskipun memiliki banyak keuntungan, implementasi sistem informasi berbasis web juga menghadapi tantangan. Salah satunya adalah masalah keamanan data. Menurut (Manuhutu & Wattimena, 2019), perlindungan terhadap data sensitif sangat penting untuk mencegah akses tidak sah dan kebocoran informasi. Selain itu, pengguna juga harus dilatih

untuk menggunakan sistem dengan efektif agar dapat memaksimalkan manfaatnya.

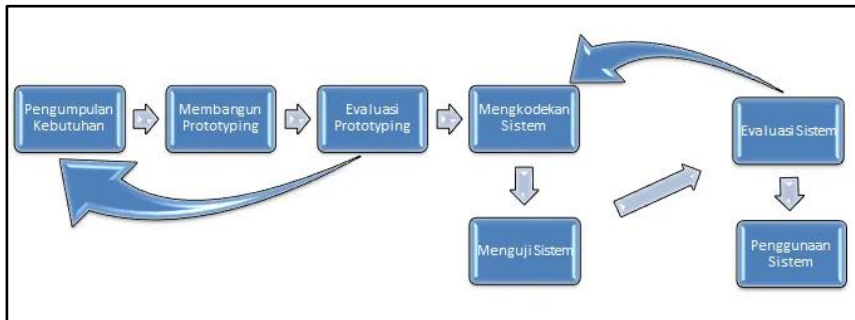
2.2.2 Sistem Informasi Akademik

Sistem informasi akademik merupakan suatu sistem yang berperan dalam pengelolaan data akademik di berbagai institusi pendidikan, baik yang bersifat formal maupun informal, mulai dari jenjang dasar hingga perguruan tinggi. Secara umum, sistem ini mencakup pengolahan berbagai data, seperti informasi tenaga pengajar, peserta didik, mata pelajaran, jadwal pembelajaran, serta data lain yang disesuaikan dengan kebutuhan setiap lembaga pendidikan. (Cahyanti & Kurnia, 2022).

Seiring dengan pesatnya perkembangan teknologi, kebutuhan akan sistem informasi yang dapat diakses secara fleksibel menjadi semakin meningkat. Sistem informasi akademik hadir sebagai solusi dalam menyediakan layanan informasi secara daring, memungkinkan pengguna untuk memperoleh data akademik dengan lebih mudah dan efisien. Sistem ini dikembangkan untuk membantu sekolah dan perguruan tinggi dalam mengelola serta menyimpan informasi akademik, sehingga mempermudah akses terhadap data pengajar, peserta didik, jadwal pembelajaran, jadwal ujian, serta berbagai data akademik lainnya yang disesuaikan dengan kebutuhan institusi pendidikan. Selain berperan sebagai pusat pengelolaan informasi, sistem informasi akademik juga berfungsi sebagai media komunikasi antara pihak-pihak yang terlibat dalam lingkungan akademik, sehingga dapat meningkatkan efektivitas dalam penyampaian informasi serta pengambilan keputusan di institusi pendidikan (Syamsul dkk., 2021).

2.2.3 Metodologi *Prototyping*

Metode *Prototyping* adalah pendekatan dalam pengembangan perangkat lunak yang melibatkan pembuatan *Model* awal sistem yang dapat berfungsi sebagai versi awal sebelum implementasi penuh. *Model* ini, atau prototipe, berperan sebagai perantara antara pengembang dan pengguna, memungkinkan keduanya untuk berinteraksi secara langsung selama proses pengembangan sistem informasi guna memastikan bahwa hasil akhir sesuai dengan kebutuhan pengguna (Mulyadi, 2023).



Gambar 2.1 Fase-Fase Metodologi *Prototyping* (Rizky, 2019)

Menurut (Rizky, 2019), adapun fase-fase dalam *Prototyping* adalah sebagai berikut:

- 1) Pengumpulan Kebutuhan
Pada tahap ini, pengembang mengidentifikasi kebutuhan perangkat lunak serta seluruh elemen sistem yang akan dikembangkan.
- 2) Membangun *Prototyping*
Pada tahap ini, pengembang membuat rancangan awal yang berfokus pada tampilan dan interaksi dengan pengguna, seperti pembuatan format *input* dan *output*.
- 3) Evaluasi *Prototyping*
Prototipe yang telah dibuat dievaluasi untuk memastikan kesesuaiannya dengan harapan dan kebutuhan pengguna.
- 4) Mengkodekan Sistem
Setelah prototipe disetujui, rancangan tersebut dikonversi ke dalam kode pemrograman agar menjadi sistem yang fungsional.
- 5) Menguji Sistem
Perangkat lunak yang telah dikembangkan diuji untuk memastikan bahwa sistem berjalan dengan baik dan sesuai dengan spesifikasi yang telah ditentukan.
- 6) Evaluasi Sistem
Setelah pengujian selesai, perangkat lunak dievaluasi oleh pengguna guna memastikan bahwa sistem telah memenuhi kebutuhan yang diharapkan.
- 7) Menggunakan Sistem
Perangkat lunak yang telah lolos uji dan disetujui oleh pengguna siap untuk digunakan dalam lingkungan operasional.

Dalam penerapannya, *Prototyping* memiliki sejumlah kelebihan dan kekurangan dalam pengembangan sistem atau perangkat lunak (Rizky, 2019). Berikut adalah rincian kelebihan dan kekurangannya:

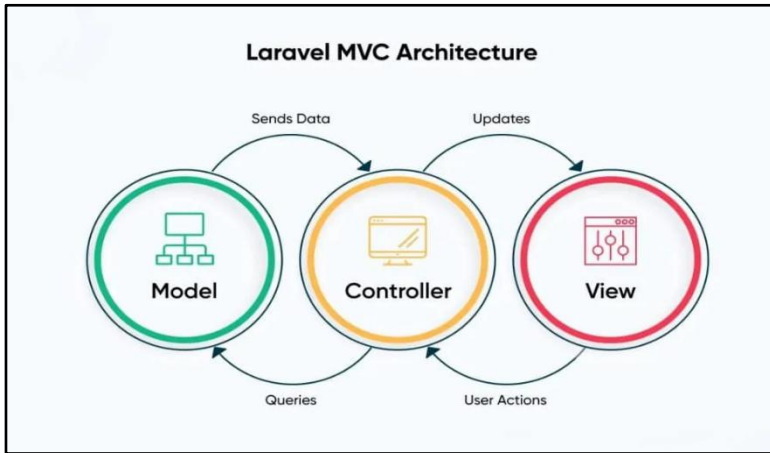
1) Kelebihan metode *Prototyping*

- a) Pengguna dapat terlibat secara langsung dalam proses pengembangan sistem, sehingga memudahkan pengembang dalam memahami dan menyesuaikan produk dengan kebutuhan pengguna.
- b) Identifikasi dan pemenuhan kebutuhan sistem menjadi lebih mudah dilakukan.
- c) Waktu yang dibutuhkan untuk mengembangkan perangkat lunak menjadi lebih singkat dibandingkan metode lain.
- d) Komunikasi antara pengembang dan pengguna menjadi lebih efektif, memungkinkan umpan balik yang lebih cepat dan akurat.
- e) Pengembang lebih mudah dalam menentukan kebutuhan spesifik pengguna berdasarkan interaksi selama proses pengembangan.
- f) Implementasi sistem menjadi lebih lancar karena pengguna memiliki pemahaman yang lebih baik tentang fungsionalitas yang diharapkan.

2) Kekurangan metode *Prototyping*

- a) Proses analisis dan perancangan dilakukan dalam waktu yang relatif singkat, yang dapat menyebabkan kurangnya perencanaan yang matang.
- b) Metode ini kurang fleksibel dalam menangani perubahan besar di tengah proses pengembangan.
- c) Meskipun pengguna dapat melihat perkembangan dari setiap versi prototipe, mereka mungkin tidak menyadari bahwa pengembangan dilakukan tanpa mempertimbangkan aspek kualitas dan pemeliharaan jangka panjang.

2.2.4 Laravel



Gambar 2.2 Arsitektur MVC Laravel

(Sumber: <https://www.rumahweb.com/journal/wp-content/uploads/2024/09/Laravel-MVC-concept.webp>)

Laravel merupakan *framework open source* berbasis PHP yang menerapkan arsitektur *Model-View-Controller* (MVC). Menurut Mulyadi, popularitas Laravel dalam beberapa tahun terakhir dipengaruhi oleh kemudahan penggunaannya serta dokumentasi yang komprehensif. Laravel menerapkan pola arsitektur *Model-View-Controller* (MVC), yang membagi aplikasi menjadi beberapa komponen, yakni manipulasi data (*Model*), pengendalian logika aplikasi (*Controller*), serta antarmuka pengguna (*View*). Penerapan pola ini memberikan keuntungan dalam hal kemudahan pemeliharaan serta skalabilitas aplikasi (Ramadhani, 2023).

Menurut (Ramadhani, 2023), arsitektur *Model-View-Controller* (MVC) terdiri dari beberapa komponen berikut:

- 1) *Model*: Merepresentasikan struktur data dalam aplikasi dan berisi fungsi-fungsi yang berhubungan dengan pengelolaan basis data, seperti penyimpanan, pembaruan, dan penghapusan data.
- 2) *View*: Bertanggung jawab dalam menampilkan antarmuka pengguna. Dalam konteks aplikasi web, *View* merupakan halaman yang disajikan kepada pengguna.
- 3) *Controller*: Berperan sebagai penghubung antara *Model* dan *View*. *Controller* menerima permintaan dari pengguna, kemudian mengolahnya dengan *Model*, dan mengirimkan hasilnya ke *View* untuk ditampilkan.

2.2.5 Black-box Testing

Black-box Testing adalah metode pengujian yang menitikberatkan pada aspek fungsional perangkat lunak, di mana penguji menentukan berbagai kondisi *input* dan melakukan pengujian berdasarkan spesifikasi fungsi dari program tersebut (Hidayat & Muttaqin, 2018).

1) Prinsip Dasar *Black-box Testing*

Black-box Testing menggunakan pendekatan "*test-to-requirements*", di mana pengujian dilakukan berdasarkan spesifikasi kebutuhan sistem. Beberapa prinsip utama dari metode ini adalah:

- a) Berfokus pada fungsionalitas sistem: Pengujian dilakukan untuk memverifikasi apakah fitur yang diuji memberikan hasil yang benar.
- b) Tidak membutuhkan pemahaman tentang kode program: Penguji hanya perlu memahami bagaimana sistem seharusnya berfungsi dari sudut pandang pengguna.
- c) Berbasis skenario pengujian: Pengujian dilakukan dengan mengembangkan skenario yang mencakup berbagai kondisi *input*, termasuk *input valid*, *invalid*, dan batasan data.

2) Jenis Teknik *Black-box Testing*

Menurut (Praniffa dkk., 2023), ada beberapa teknik yang umum digunakan dalam *Black-box Testing*, antara lain:

- a) *Equivalence Partitioning*: Membagi domain *input* menjadi beberapa kelas yang mewakili data *valid* dan *invalid*.
- b) *Boundary Value Analysis*: Menguji batas nilai minimum dan maksimum dari *input* yang diterima sistem.
- c) *Decision Table Testing*: Mengidentifikasi kombinasi *input* dan memverifikasi hasilnya.
- d) *State Transition Testing*: Menguji perubahan status sistem berdasarkan masukan tertentu.

3) Keunggulan dan Kekurangan

Black-box Testing memiliki kelebihan dan kekurangan dalam penerapannya. Salah satu keunggulannya adalah kemampuannya untuk mengidentifikasi aspek-aspek yang tidak sesuai dengan spesifikasi kebutuhan yang telah ditentukan dalam pengembangan perangkat lunak. Namun, kelemahan metode ini terletak pada keterbatasan pengetahuan penguji tentang perangkat lunak yang diuji, sehingga pengujian tidak dapat dilakukan secara menyeluruh (Praniffa dkk., 2023).

2.2.6 User Acceptance Testing

User Acceptance Testing (UAT) merupakan tahap akhir dalam proses pengujian perangkat lunak yang bertujuan untuk memastikan bahwa sistem yang dikembangkan dapat berfungsi sesuai dengan kebutuhan dan spesifikasinya. Pengujian ini dilakukan oleh pengguna akhir menggunakan pendekatan *black-box testing* guna menilai apakah sistem telah memenuhi persyaratan yang telah ditetapkan. UAT juga berfungsi sebagai bukti bahwa perangkat lunak mampu mendukung kebutuhan pengguna secara efektif sebelum diterapkan dalam lingkungan operasional. Sebagai salah satu prosedur penting dalam pengembangan perangkat lunak, UAT memastikan bahwa sistem yang diuji tidak hanya sesuai dengan spesifikasi teknis, tetapi juga memenuhi kebutuhan nyata pengguna. Oleh karena itu, tahap ini sering disebut sebagai pengujian beta, pengujian aplikasi, atau pengujian pengguna akhir (Wahyudi dkk., 2023).

UAT terdiri dari beberapa jenis pengujian, termasuk Alpha dan Beta *testing*, yang memiliki peran berbeda dalam menilai kesiapan perangkat lunak. Pengujian Alpha dilakukan secara internal oleh pengembang untuk mendeteksi *bug* sejak tahap awal pengembangan, sering kali menggunakan metode *black-box testing* guna menguji fungsionalitas sistem. Sebaliknya, pengujian Beta melibatkan pengguna eksternal untuk mendapatkan umpan balik dari pengalaman penggunaan nyata sebelum perangkat lunak dirilis. Pada tahap Beta, kuesioner dapat digunakan sebagai alat evaluasi guna mengukur tingkat penerimaan pengguna terhadap aplikasi yang diuji, sehingga pengembang dapat melakukan perbaikan lebih lanjut sebelum peluncuran resmi (Yakub dkk., 2024).

2.2.7 Usability Testing

Usability merupakan salah satu atribut kualitas yang digunakan untuk menilai tingkat kemudahan suatu antarmuka dalam digunakan oleh pengguna. Pengujian *usability* (*usability testing*) bertujuan untuk mengevaluasi sejauh mana sistem dapat digunakan secara efektif, efisien, dan memuaskan oleh pengguna (Fatikha & Wahyudi, 2025). Terdapat 5 komponen utama dalam *usability*, yaitu sebagai berikut:

- 1) *Learnability*, yaitu aspek yang mengukur tingkat kemudahan pengguna dalam memahami dan menggunakan sistem pada saat pertama kali berinteraksi dengan sistem tersebut.
- 2) *Efficiency*, yaitu aspek yang mengukur tingkat kecepatan dan ketepatan pengguna dalam menyelesaikan tugas setelah memahami cara kerja sistem.

- 3) *Memorability*, yaitu aspek yang mengukur kemampuan pengguna dalam mengingat kembali cara penggunaan sistem setelah tidak menggunakannya dalam jangka waktu tertentu.
- 4) *Errors*, yaitu aspek yang mengukur jumlah kesalahan yang dilakukan pengguna selama menggunakan sistem, tingkat keparahan kesalahan tersebut, serta kemudahan pengguna dalam memperbaiki kesalahan yang terjadi.
- 5) *Satisfaction*, yaitu aspek yang mengukur tingkat kepuasan pengguna terhadap tampilan antarmuka dan kenyamanan sistem secara keseluruhan saat digunakan.