BAB 2
TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori berisikan tentang teori yang berhubungan dengan proyek akhir yang digunakan sebagai arahan dalam memecahkan masalah proyek akhir.

2.1.1 Raudhatul Athfal Fadhilah

Raudhatul Athfal Fadhilah merupakan salah satu lembaga pendidikan anak usia dini yang berdiri pada 11 Januari 2009 yang berlokasi di Jl. Muhajirin, Sidomulyo Barat, Kecamatan Tampan, Kota Pekanbaru, Provinsi Riau. Sekolah ini berperan sebagai tempat pendidikan dasar bagi anak-anak usia dini sebelum melanjutkan ke jenjang pendidikan dasar formal. Fokus utama dari RA Fadhilah adalah memberikan pendidikan yang seimbang antara aspek religius, moral, sosial, dan akademik, guna membentuk karakter anak yang mandiri, berakhlak mulia, dan siap menghadapi tahap pendidikan berikutnya.

Gambar 2. 1 Raudhatul Athfal Fadhilah

Sebagai lembaga pendidikan formal, RA Fadhilah turut menyelenggarakan program Penerimaan Peserta Didik Baru (PPDB) yang bertujuan untuk menyeleksi dan menerima siswa baru secara sistematis. Dengan meningkatnya kebutuhan informasi yang cepat dan aksesibel di era digital, maka penting bagi RA Fadhilah untuk memiliki sistem informasi berbasis *web* yang dapat menunjang penyampaian

informasi profil sekolah serta proses pendaftaran secara *online*. Penerapan teknologi informasi di lingkungan RA tidak hanya meningkatkan efisiensi, namun juga mendukung transparansi dan pelayanan yang lebih baik kepada masyarakat.

2.1.2 Sistem Informasi

Menurut (Rahmawati & Sumarno, 2020), Sistem informasi adalah kombinasi terorganisasi apapun dari manusia, perangkat lunak, perangkat keras, jaringan komunikasi, sumber data dan kebijakan serta prosedur terorganisasi yang menyimpan, mengambil, mengubah, dan memisahkan informasi dalam sebuah organisasi.

Sistem informasi memainkan peran penting dalam mendukung kegiatan bisnis operasional, membantu manajemen dalam pengambilan keputusan, serta memperkuat keunggulan strategi kompetitif organisasi. Sebagai contoh, sebuah perusahaan dapat mencapai keunggulan strategis dengan mengubah seluruh data mereka ke dalam basis data yang dapat diakses melalui alat penghubung standar, seperti *browser web* ataupun aplikasi *mobile* (Rahmawati & Sumarno, 2020).

2.1.3 PHP

Hypertext Preprocessor (PHP) adalah bahasa *scripting* dan *interpreter* yang bebas untuk digunakan dan secara luas dipakai untuk pengembangan *web*. Kegunaan utama dari bahasa pemrograman ini adalah sebagai *server-side scripting,* tapi juga bisa digunakan sebagai *command-line scripting.* Pada awalnya, Rasmus Lerdorf sebagai *founder* mengembangkan *PHP* dengan nama *Personal Home Page* (Sheldon, 2023).

Menurut (Tim, 2014), salah satu keuntungan *PHP* adalah kemudahan integrasinya dengan *database*. *PHP* dapat langsung mendukung berbagai *database* tanpa perlu menginstal konektor seperti pada bahasa pemrograman *Java*, sehingga menjadikannya sangat fleksibel dalam berinteraksi dengan berbagai *database*. Untuk menghubungkan *PHP* dengan *database*, pengguna hanya perlu mengetahui nama *database*, lokasinya, serta *username* dan *password* yang diperlukan untuk mengakses *database* tersebut. Berikut beberapa keuntungan menggunakan *PHP*:

1) Pengguna tidak perlu membayar untuk menggunakan *PHP*.
2) Mudah digunakan, dokumentasi *PHP* banyak ditemukan di *internet*.

3) Dapat dijalankan di berbagai sistem operasi seperti *Windows, Linux, MacOS.*
4) Mendukung banyak *database.*

2.1.4 Web Development Framework

Menurut (Sheldon, 2023), *Web Development Framework* adalah sekumpulan *library, resources* dan *tools* yang digunakan untuk membangun dan mengelola aplikasi *web*, layanan *web*, serta mengembangkan *Application Programming Interface (API)*. *Framework* memiliki beberapa fungsi utama dalam pengembangan aplikasi atau sistem, seperti mempercepat proses pengembangan dan menjaga konsistensi dalam kode program. Selain itu, *framework* juga meningkatkan kualitas aplikasi atau sistem serta mempermudah pemeliharaannya. *Framework* tersebut dibagi atas 2 kelompok, yaitu *front-end frameworks dan back-end frameworks.* Contoh *front-end framework* meliputi *Angular.JS, React.JS, Vue,* dan *Next.JS,* sedangkan contoh *back-end framework* adalah *Laravel, Django*, dan *Flask*.

(Shetty dkk., 2020) membahas bahwa *Front-end frameworks* bertujuan sebagai bagian dari aplikasi atau situs *web* yang berinteraksi langsung dengan pengguna. *Framework* ini mencakup semua elemen yang dihadapi pengguna secara langsung, seperti gambar, tombol, warna teks, diagram dan tabel. *Developer* harus memastikan bahwa aplikasi tersebut responsif, seperti contoh posisi komponen ditampilkan dengan benar disemua ukuran perangkat. Sedangkan *back-end frameworks* menurut (Shetty dkk., 2020) adalah bagian dari *web development* yang melibatkan berbagai aktifitas krusial, seperti menjaga *API* dari ancaman pihak luar, *autentikasi users*, memastikan bahwa interaksi dengan *database* berjalan lancar, dan memproses permintaan dari *users* serta menampilkan informasi yang sesuai. *Back-end frameworks* mempermudah tugas—tugas *developer*, membuat proses menjadi lebih efisien dan bebas dari kerepotan.

2.1.5 Laravel

Laravel adalah *framework model, view* dan *controller (MVC)* berbasis *PHP* yang dibuat oleh Taylor Otwell pada tahun 2011 untuk menggantikan peran *Codeigniter* yang dia anggap terlalu rumit. *Laravel* dirancang untuk membantu *developer* membangun aplikasi berbasis *web* yang aman dan kuat. *Framework* ini

menyediakan berbagai *package* dan *API* yang mempermudah integrasi berbagai komponen dalam aplikasi, sehingga membuat proses pengembangan menjadi lebih efisien. *Laravel* menggunakan *PHP Composer* untuk mengelola dependensi, menginstal paket, dan menjalankan proyek, memungkinkan pengembang menggunakan perintah artisan untuk dengan mudah membuat *model, controller,* dan layanan. Selain itu, *Laravel* mendukung teknologi *front-end* seperti *Bootstrap* dan *front-end framework* lainnya, memberikan fleksibilitas kepada pengembang dalam merancang antarmuka pengguna sesuai preferensi mereka (Bagwan & Ghule, 2019).

Proyek *Laravel* beroperasi dalam struktur direktori *root* dan dapat diinstal dengan mudah melalui perintah *Composer. Laravel* dilengkapi dengan beberapa paket yang telah tersedia dan mudah diakses secara *online*. *Model* di *Laravel* berfungsi sebagai wadah untuk menyimpan *kredensial database* dan menyederhanakan penyimpanan serta manajemen hubungan menggunakan fitur *query builder* dan *Eloquent ORM. Controller* berperan sebagai jembatan antara tampilan antarmuka pengguna (UI) dan *database*, menangani *query* dan logika. *View* berisi bagian *HTML* dari proyek, menjadi inti dari sistem karena merupakan titik utama interaksi dengan pengguna. *Laravel* juga menyediakan kemampuan *routing*, memungkinkan pengembang mendefinisikan *URL* utama yang akan diakses oleh pengguna di *web* (Bagwan & Ghule, 2019).

2.1.6 Database Server (MySQL)

Database diambil dari kata data yang berarti catatan fakta, sedangkan *base* dapat diartikan sebagai pusat berkumpulnya data tersebut. Sehingga *database* dapat diartikan sebagai kumpulan data yang saling berhubungan, disimpan di perangkat keras dan dikelola oleh perangkat lunak untuk dimanipulasi (Wahyudi dkk., 2022). Terdapat 2 (dua) jenis *database* saat ini, yaitu *Structured Query Language (SQL)* dan *Not Only SQL (NoSQL). SQL* adalah bahasa yang digunakan untuk mengelola dan memanipulasi data dalam *Relational Database Management System (RDBMS). Database SQL* menyimpan data dalam tabel dengan skema terstruktur, di mana hubungan antar tabel ditentukan melalui *primary key* dan *foreign key*.

SQL digunakan dalam sistem *database* seperti *MySQL, PostgreSQL*, dan *Microsoft SQL Server.* Sedangkan *Not Only SQL (NoSQL)* adalah istilah yang

merujuk pada *jenis database non-relational* yang dirancang untuk menyimpan, mengelola, dan mengambil data yang tidak memerlukan skema tetap. *NoSQL* cocok untuk data yang besar, tidak terstruktur, atau semi-terstruktur, dan sering digunakan dalam aplikasi *modern* seperti *big data* dan media sosial. Contoh basis data *NoSQL* termasuk *MongoDB, Cassandra,* dan *Redis* (Khan dkk., 2022). *MySQL* dikembangkan oleh Michael "Monty" Wideners, David Axmark dan Allan Larsson pada tahun 1979. *MySQL* adalah sebuah *RDBMS* yang *multithread, multi-user* dengan jumlah instalasi mencapai 6 juta diseluruh dunia. *Database* ini di distribusikan secara gratis dibawah *lisensi GPL (General Public License),* sehingga setiap pengguna dapat dengan bebas menggunakan *MySQL* dengan keterbatasan tidak boleh digunakan untuk tujuan komersil.

2.1.7 API

Application Programming Interface (API) merupakan seperangkat protokol, fungsi, dan definisi yang memungkinkan satu aplikasi berkomunikasi dengan aplikasi lainnya, tanpa perlu mengetahui bagaimana aplikasi tersebut diimplementasikan secara internal. Salah satu bentuk *API* yang paling umum digunakan saat ini adalah *RESTFUL API*, yang berbasis pada arsitektur *Representational State Transfer* dan menggunakan metode *HTTP* standar seperti *GET, POST, PUT,* dan *DELETE* untuk berinteraksi dengan data. *API* memegang peran penting dalam pengembangan perangkat lunak modern karena mampu meningkatkan efisiensi integrasi antar sistem, modularitas, serta kemudahan dalam pengelolaan data secara terpusat. Menurut penelitian oleh Haupt et al. (2021), desain *API* yang baik dapat secara signifikan meningkatkan kualitas perangkat lunak dari aspek *usability, maintainability,* hingga *compatibility*.

Dalam praktiknya, *API* sangat berperan dalam pengembangan aplikasi berbasis *web* maupun *mobile*. Misalnya, penelitian oleh Dewi dan Nasution (2024) menunjukkan bahwa implementasi *REST API* dalam sistem *e-learning* di Unhar Medan mempermudah integrasi lintas platform dan mempercepat proses pengelolaan data akademik. Selain itu, studi oleh Pramudito (2025) dalam pengembangan aplikasi *petshop berbasis web* membuktikan bahwa penggunaan *REST API* mampu menurunkan konsumsi sumber daya sistem serta meningkatkan performa aplikasi secara keseluruhan. Seiring dengan adopsi arsitektur mikroservis,

API juga berfungsi sebagai titik utama integrasi antar-layanan, dan evolusinya perlu dirancang dengan cermat untuk menjaga kompatibilitas serta menghindari gangguan sistem (Zhou et al., 2023). Oleh karena itu, *API* bukan hanya alat teknis, tetapi juga elemen strategis dalam membangun sistem informasi yang efisien, fleksibel, dan berkelanjutan.

2.1.8 Payment Gateway Midtrans

Payment Gateway adalah gerbang atau alat transaksi yang disediakan oleh sebuah layanan yang memungkinkan pemrosesan kartu kredit atau pembayaran langsung kepada pelanggan yang menjalankan bisnis elektronik atau *online* (Arsana & Ramadhani, 2020). Merujuk pada Peraturan Bank Indonesia Nomor 18/40/PBI/2016 tentang Penyelenggaran Pemrosesan Transaksi Pembayaran, *Payment gateway* merupakan sebuah layanan elektronik yang digunakan dalam melakukan proses transaksi pembayaran dengan menggunakan alat pembayaran dengan media kartu, uang elektronik, dan/atau *Propreitary Channel*.

Midtrans merupakan salah satu platform untuk melakukan *Payment Gateway*. *Midtrans* merupakan sistem pembayaran yang dapat memfasilitasi penjual dan pembeli untuk melakukan transaksi. *Midtrans* menyediakan *tools* terintegrasi sesuai kebutuhan pembayaran secara *online* dengan kartu kredit, bahkan penarikan uang dan pengiriman uang. Dengan *midtrans* dapat melakukan pembayaran belanja *online*, donasi, biaya sekolah, produk berlangganan dan penarikan uang dengan mudah, cepat dan aman. (Nisrina, Y. E. Dkk., 2019).

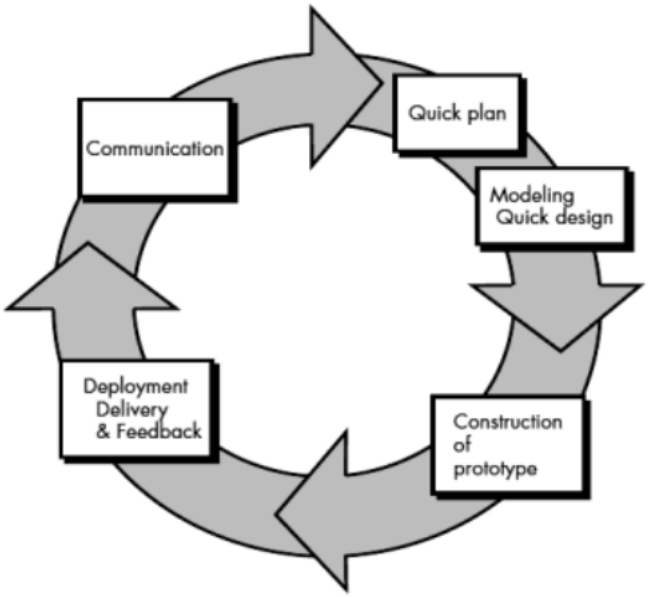
Pada penelitian terdahulu yang dilakukan oleh (Hariselmi, dkk. 2022) yang berjudul ”Implementasi Sistem Informasi Pembayaran SPP Berbasis Android Dengan *Payment Gateway Midtrans*”. Penelitian ini memberikan kemudahan dalam pembayaran SPP dengan mengintegrasikan *Midtrans* sebagai pembayaran *online* dan dapat dijalankan secara *realtime*.

2.1.9 Prototype

Menurut Rizky (2019), *Prototyping* perangkat lunak adalah salah satu metode siklus hidup sistem yang didasarkan pada konsep *model* bekerja (*working model*). Tujuannya adalah mengembangkan *model* menjadi sistem final. Artinya sistem akan dikembangkan lebih cepat dari pada metode tradisional dan biayanya menjadi lebih 24 rendah. Ada banyak cara untuk melakukan *prototyping*, begitu pula dengan

penggunaannya.

Metode *Prototype* menurut Pressman (2002:40), dimulai dengan mengumpulkan kebutuhan. Pengembang dan klien bertemu guna mendefinisikan *obyektif* keseluruhan dari perangkat lunak, mengidentifikasikan segala kebutuhan dari segi *input* dan format *output* serta gambaran *interface*, kemudian dilakukan perancangan cepat. Dari hasil perancangan cepat tersebut nantinya akan dilakukan pengujian dan evaluasi. Penjelasan lengkap pada metode *prototype* akan dijelaskan melalui gambar pada halaman selanjutnya.



Gambar 2. 2 Metode *Prototype*

Pada Gambar 2.2 di atas terdapat tiga siklus yang akan dijelaskan sebagai berikut:

a) *Communication*

Tahap communication merupakan tahap awal untuk mengidentifikasi kebutuhan dan permasalahan pengguna. Pada tahap ini, pengembang berkomunikasi dengan pengguna melalui wawancara, observasi, diskusi, atau pengumpulan dokumen pendukung.

Komunikasi dilakukan untuk mengetahui tujuan sistem, proses yang sedang berjalan, kendala yang dihadapi, fungsi yang dibutuhkan, serta harapan pengguna terhadap sistem. Hasil tahap ini berupa informasi awal mengenai kebutuhan fungsional dan nonfungsional sistem.

b) *Quick Plan*

Quick plan merupakan tahap penyusunan rencana awal pengembangan *prototype* secara cepat. Berdasarkan kebutuhan yang telah diperoleh, pengembang menentukan ruang lingkup sistem, fitur yang akan diprioritaskan, waktu pengerjaan, sumber daya, dan teknologi yang akan digunakan.

Perencanaan pada metode *prototyping* tidak dibuat secara terlalu rinci seperti pada *metode* pengembangan tradisional. Hal ini karena kebutuhan sistem masih dapat berubah setelah pengguna mencoba *prototype*. Hasil tahap ini berupa rencana kerja awal yang menjadi pedoman dalam membuat desain dan membangun *prototype*.

c) *Modelling Quick Design*

Pada Tahap *modeling quick design* merupakan tahap pembuatan desain awal sistem. Desain dibuat secara cepat dengan berfokus pada bagian yang dapat dilihat dan digunakan langsung oleh pengguna. Desain tersebut dapat berupa rancangan antarmuka, struktur menu, alur proses, *model data*, diagram sistem, serta hubungan antara pengguna dan fungsi-fungsi yang tersedia.

Tujuan utama tahap ini adalah memberikan gambaran awal mengenai bagaimana sistem akan bekerja. Desain yang dibuat pada tahap ini belum bersifat final. Desain masih dapat mengalami perubahan berdasarkan hasil evaluasi dan masukan dari pengguna.

d) *Construction of Prototype*

Pada Tahap *construction of prototype* merupakan tahap pembangunan *prototype* berdasarkan rancangan awal yang telah dibuat. Pada tahap ini, desain diterjemahkan menjadi bentuk sistem yang dapat dijalankan dan diuji oleh pengguna.

Prototype tidak selalu memuat seluruh fungsi sistem secara lengkap. *Prototype* dapat hanya menampilkan fungsi utama, alur kerja, atau antarmuka yang diperlukan untuk membantu pengguna memahami sistem yang sedang dikembangkan. Tujuan pembangunan *prototype* bukan langsung menghasilkan produk akhir, melainkan menyediakan

model sistem yang dapat digunakan untuk mendapatkan umpan balik dari pengguna.

e) *Deployment, Delivery, and Feedback*

Pada Tahap *deployment, delivery, and feedback* merupakan tahap penyerahan *prototype* kepada pengguna untuk dicoba dan dievaluasi. Pengguna menggunakan *prototype* untuk menilai apakah fungsi, tampilan, dan alur sistem telah sesuai dengan kebutuhan.

Pada tahap ini, pengguna dapat memberikan masukan berupa perubahan tampilan, perbaikan proses, penambahan fungsi, pengurangan fungsi, atau penyederhanaan alur sistem. Pengembang kemudian mencatat dan menganalisis setiap masukan yang diberikan.

Apabila *prototype* belum sesuai dengan kebutuhan, proses pengembangan kembali ke tahap komunikasi. Pengembang dan pengguna mendiskusikan kembali kebutuhan serta perubahan yang harus dilakukan.

Dengan demikian, metode *prototyping* menekankan keterlibatan aktif pengguna selama proses pengembangan. Pengguna tidak hanya memberikan kebutuhan pada awal pengembangan, tetapi juga ikut mengevaluasi dan mengarahkan perbaikan sistem pada setiap siklus.

2.1.10 Chatbot

Chatbot adalah sebuah program komputer yang bertujuan unuk mensimulasikan sebuah kecerdasan buatan bertujuan untuk melakukan sebuah percakapan dengan manusia (Shawar & Atwell, 2007). *Chatbot* merupakan hasil implementasi dari bidang ilmu pengolahan bahasa alami, pembelajaran mesin, rekayasa perangkat lunak dan kecerdasan buatan. *Chatbot* dirancang untuk mensimulasikan percakapan dengan manusia menggunakan aturan atau kecerdasan buatan melalui antarmuka percakapan melalui teks tertulis atau lisan.

Sebuah *chatbot* dapat berjalan menggunakan pembelajaran mesin melalui kecerdasan buatan untuk menangkap pola percakapan yang memungkinkannya untuk meniru percakapan manusia dan bereaksi terhadap permintaan tertulis atau lisan sehingga *chatbot* mampu merespon dengan balasan yang sesuai untuk memberikan layanan atau informasi. Saat ini *AI* dalam bentuk *chatbot* banyak

digunakan dalam layanan pelanggan. *Chatbot* dapat memberikan jawaban cepat dan akurat sesuai dengan pertanyaan pelanggan, sehingga diharapkan dapat meningkatkan juga kepuasaan pelanggan. Contoh penerapannya adalah rancang bangun *chatbot* untuk meningkatkan performa bisnis yang bertujuan untuk menambahkan fungsi dan juga meningkatkan performa bisnis (Amalia & Wibowo, 2019).

2.1.11 Ollama

Ollama merupakan *platform open-source* yang memungkinkan pengguna menjalankan *Large Language Model (LLM*) secara lokal pada perangkat tanpa memerlukan layanan *cloud*. *Ollama* menyediakan lingkungan yang sederhana untuk mengunduh, mengelola, dan menjalankan berbagai *model* bahasa besar seperti *Llama*, *Mistral*, dan *Gemma*, serta mendukung integrasi melalui *API* untuk pengembangan aplikasi berbasis kecerdasan buatan (Lin & Safi, 2025).

Menurut Riza et al. (2025), penggunaan *Ollama* memungkinkan implementasi model bahasa besar secara mandiri pada lingkungan lokal sehingga meningkatkan privasi data, mengurangi ketergantungan pada layanan pihak ketiga, serta mendukung pengembangan sistem *AI* yang lebih aman dan efisien. Oleh karena itu, *Ollama* banyak digunakan sebagai *runtime engine* yang bertugas memproses masukan pengguna dan menghasilkan respons berdasarkan *model* bahasa yang dijalankan.

Pada penelitian ini, *Ollama* digunakan sebagai *platform* untuk menjalankan *model Large Language Model (LLM*) yang berfungsi sebagai *chatbot* informasi PPDB. Ollama menerima pertanyaan dari pengguna, memprosesnya menggunakan *model* yang telah dikonfigurasi, dan menghasilkan jawaban yang sesuai dengan informasi yang tersedia pada sistem.

2.1.12 Black Box Testing

Black box testing atau biasa disebut *behavioural testing* adalah *metode* pengujian perangkat lunak berdasarkan perilakunya, bukan menguji struktur atau kode internal dari suatu program dan berfokus pada *input* dan *output* yang telah dicapai (Tiwari dkk, 2017). Menurut (Nidhra, S., & Dondeti, J. 2012). *Black box testing* merupakan pengujian yang dilakukan berdasarkan spesifikasi kebutuhan dan tidak memerlukan pemeriksaan kode pada aplikasi. Proses ini murni dilakukan

berdasarkan sudut pandang pengguna. *Black box testing* berperan penting dalam proses pengujian perangkat lunak untuk membantu validasi fungsionalitas dari sistem. Kelebihan dari *metode* ini adalah penguji tidak perlu memiliki pengetahuan khusus mengenai bahasa pemrograman dan juga pengetahuan tentang implementasi.

2.1.13 Fungsionalitas Testing

Menurut Enstein dkk. (2022) Uji fungsionalitas merupakan salah satu tahapan yang penting dalam pengembangan sebuah sistem, dimana dalam tahapan ini akan diuji keseluruhan fungsionalitas dan kinerja sistem sebelum dirilis kepada pengguna. Uji fungsionalitas akan membantu mengidentifikasi dan memperbaiki kesalahan, *bug* atau masalah fungsi lainnya yang ada didalam sistem. Dengan melakukan pengujian terhadap fitur dan aspek *web*, pengembang dapat memastikan bahwa *web* yang dikembangkan telah berjalan sesuai dengan standar maupun kualitas yang di inginkan. Adanya uji *fungsionalitas* tentu akan sangat membantu dalam mengidentifikasi area yang memerlukan perbaikan. Hal ini dikarenakan pengujian berfokus pada fitur dan fungsionalitas sistem yang dapat dilihat dan ditinjau oleh pengguna (Kustiawan, Cholifah, Destriana, & Heriyani, 2022).

Salah satu *metode* yang dapat digunakan dalam menguji sistem ini adalah *black box testing*. *Metode* ini akan diuji melalui sudut pandang pengguna ekternal yang tidak memerlukan akses terhadap kode sumber atau struktur internal dari sistem, sehingga sangat memungkinkan sekali pengguna yang tidak pernah terlibat sama sekali dalam pengembangan sistem untuk melakukan pengujian (Novalia & Voutama, 2022). Pengujian menggunakan *metode black box testing* dapat membantu mengidentifikasi masalah ataupun kesalahan yang mungkin terjadi ketika pengguna menggunakan sistem, hal ini dikarenakan identifikasi masalah dilakukan sesuai dengan persefektif pengguna.

Selain itu dengan ketika menguji menggunakan *metode black box testing* akan sangat membantu pengembang dalam memastikan *User Interface (UI)* dapat digunakan dengan mudah dan *responsive* sehingga mampu memberikan pengalaman yang menyenangkan bagi pengguna. Pengujian menggunakan *black box testing* akan membantu memastikan bahwa sistem telah memenuhi persyaratan fungsional sesuai dengan kebutuhan dan spesifikasi yang telah di tentukan,

sehingga sistem dapat menjalankan fungsi sesuai dengan harapan pengguna (Nugraha, Ariyana, & Nurnawati, 2022).

Pengujian *blackbox* berdasarkan pada spesifikasi fungsional dan persyaratan yang telah ditetapkan di awal. Penguji hanya berfokus pada keluaran dari sistem dan perilaku (kinerja) sistem tanpa memeriksa bagaimana algoritma atau kode program sistem. Uji coba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori yaitu:

1) Fungsi yang salah atau tidak ada
2) Kesalahan dalam struktur dan akses basis data
3) Kesalahan desain antarmuka
4) Kesalahan perilaku sistem
5) Kesalahan saat inisialisasi atau terminasi

2.1.14 Usability Testing

Usability testing merupakan proses pengujian kelayakan sebuah *website* yang bertujuan untuk memastikan *website* tersebut mudah untuk digunakan, efektif, dan efisien sebuah produk atau sistem yang akan berpengaruh terhadap kepuasan pengguna (Wicaksono, 2023). *Usability testing* merupakan proses pengujian kelayakan sebuah *website* yang bertujuan untuk memastikan *website* tersebut mudah untuk digunakan, efektif, dan efisien bagi pemustaka. Pengujian kelayakan dari sebuah *website* terdapat beberapa *metode* yang digunakan, salah satunya adalah *metode* yang dikembangkan oleh Jacob Nielsen. Konsep dari Jacob Nielsen dipilih karena lima atribut ketergunaan yang diteliti merupakan cerminan dari kebutuhan pemustaka.

Pemilihan *usability testing* karena struktur dari karakteristik kualitasnya lebih dapat diterima dan dijelaskan dengan sistematis serta tempat pengunaan ditentukan dengan jelas (Benmoussa et al., 2019). *Usability testing* terdiri dari lima variabel yang terdiri dari *learnability, efficiency, memorability, errors,* dan *satisfaction* (Supriyatna, 2019). Lima variabel ketergunaan atau *usability testing* yang terdiri dari: 1) *Learnability* untuk melihat kemudahan dalam menyelesaikan tugas; 2) Efisiensi untuk penghematan waktu; 3) *Memorability* untuk melihat seberapa mudah *website* mudah untuk diingat; 4) *Errors* untuk melihat seberapa tingkat kesalahan ketika mengakses *website*; 5) *Satisfaction* untuk melihat kepuasan

pemustaka dalam menggunakan *website*.

Berikut adalah beberapa langkah penting dalam melakukan pengujian *usability*:

1. Persiapan

   Pada tahap ini, pengembang perangkat lunak harus mempersiapkan dokumen dokumen yang diperlukan untuk melakukan pengujian *usability*, seperti skenario pengujian, kuesioner, dan catatan pengamatan.

2. Identifikasi masalah *usability*

   Langkah selanjutnya adalah mengidentifikasi masalah *usability* yang terdapat pada perangkat lunak. Hal ini dapat dilakukan dengan cara melakukan observasi langsung terhadap pengguna saat menggunakan perangkat lunak, mengumpulkan *feedback* dari pengguna, atau menggunakan alat bantu seperti *eye-tracking* dan pengukuran waktu respons.

3. Analisis Data

   Analisis data ini dapat membantu pengembang perangkat lunak untuk mengidentifikasi masalah-masalah yang terdapat pada perangkat lunak dan menentukan tindakan perbaikan yang perlu dilakukan.

4. Perbaikan perangkat lunak

   Perbaikan ini harus dilakukan dengan memperhatikan kebutuhan dan keinginan pengguna serta prinsip-prinsip desain antarmuka pengguna yang baik.

5. Pengujian ulang

   Langkah selanjutnya adalah melakukan pengujian ulang untuk memastikan bahwa masalah-masalah *usability* telah berhasil diperbaiki. Pengujian ulang dapat dilakukan dengan menggunakan teknik yang sama dengan pengujian sebelumnya atau menggunakan teknik pengujian yang berbeda untuk memperoleh hasil pengujian yang lebih akurat.

6. Evaluasi Akhir

   Pada tahap ini, pengembang perangkat lunak akan mengevaluasi hasil

pengujian dan memastikan bahwa perangkat lunak sudah memenuhi standar *usability* yang ditetapkan. Selain itu, pengembang perangkat lunak juga harus memastikan bahwa perangkat lunak sudah memenuhi kebutuhan dan harapan pengguna. (Wicaksono, 2023).

2.1.15 Integration Testing

Integration Testing merupakan pengujian yang dilakukan untuk memastikan bahwa komponen atau modul yang telah digabungkan dapat saling berinteraksi dan bekerja sesuai dengan fungsi yang diharapkan. Menurut Hellhake et al. (2022), *integration testing* merupakan tahap penting dalam pengujian perangkat lunak untuk memverifikasi perilaku sistem setelah beberapa komponen diintegrasikan. Pengujian ini tidak hanya berfokus pada fungsi masing-masing komponen secara terpisah, tetapi juga pada hubungan dan komunikasi yang terjadi antar komponen.

Menurut Ding et al. (2023), *integration testing* digunakan untuk membantu memastikan operasi modul perangkat lunak tetap benar dan stabil setelah dilakukan integrasi. Pada proses pengujian ini, hubungan antar bagian sistem diperiksa untuk menemukan kesalahan yang mungkin tidak terlihat ketika setiap komponen diuji secara terpisah. Oleh karena itu, *integration testing* diperlukan terutama pada sistem yang memiliki beberapa modul atau berkomunikasi dengan layanan lainnya.

Pada proyek akhir ini, *Integration Testing* digunakan untuk menguji komunikasi antara Sistem Informasi PPDB RA Fadhilah dengan layanan eksternal yang digunakan, yaitu *Ollama* pada *chatbot*, *Fonnte* pada layanan notifikasi *WhatsApp*, serta *Midtrans* sebagai *payment gateway*. Pengujian dilakukan untuk memastikan proses pengiriman *request*, pemrosesan oleh layanan terkait, penerimaan *response*, serta pembaruan data pada sistem dapat berjalan sesuai dengan alur integrasi yang telah dirancang.

2.2 Penelitian Terdahulu

Penelitian terdahulu digunakan sebagai referensi dan masukan dalam mendukung penelitian yang dilakukan. Adapun penelitian terdahulu yang digunakan pada proyek akhir ini adalah sebagai berikut:

Penelitian pertama (Khasbulloh & Karim, 2023) ” Rancang Bangun Sistem Informasi Penerimaan Peserta Didik Baru Berbasis *Web* menggunakan *Framework*

Laravel". Bertujuan merancang sistem informasi penerimaan peserta didik baru berbasis *web* pada Pondok Pesantren Bahrul Maghfiroh yang sebelumnya menggunakan proses manual yang menyulitkan calon santri dari luar daerah. Sistem ini dikembangkan dengan *metode Rapid Application Development* (*RAD*), yang dinilai efisien dalam mempercepat proses pembangunan perangkat lunak, serta menggunakan *framework Laravel* berbasis *PHP* dan basis data *MySQL*. Fitur-fitur fungsional mencakup kemampuan admin untuk mengelola data peserta, memverifikasi pendaftaran, dan memberikan pengumuman kelulusan, sedangkan calon peserta dapat mendaftar secara daring, mengunggah dokumen, dan mencetak formulir serta bukti kelulusan. Pengujian sistem menggunakan *metode black box* menunjukkan hasil yang valid dan sesuai kebutuhan. Selain itu, antarmuka sistem dirancang agar sederhana dan nyaman digunakan. Penelitian ini mendukung pentingnya digitalisasi proses administrasi pendidikan untuk meningkatkan efisiensi dan aksesibilitas, terutama di lembaga pesantren yang masih banyak bergantung pada *metode* konvensional. Sistem ini berhasil menyederhanakan proses administrasi dan mempercepat alur informasi antara pihak pesantren dan calon peserta didik.

Penelitian kedua oleh (Melati dkk., 2024) dengan judul "Rancang Bangun Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) Pada SMK Muhammadiyah Salawati Berbasis *Website* Menggunakan *Metode Waterfall*". Mengembangkan Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) berbasis *web* untuk SMK Muhammadiyah Salawati dengan *metode Waterfall*, menggunakan teknologi *PHP*, *MySQL, HTML,* dan *CSS.* Sistem ini dikembangkan untuk menjawab kebutuhan akan efisiensi dalam proses penerimaan siswa, khususnya bagi sekolah dengan akses geografis yang sulit namun memiliki infrastruktur jaringan *internet* yang memadai. Melalui pendekatan *Research and Development (R&D),* penelitian ini melakukan serangkaian validasi oleh ahli materi dan media serta uji coba kepada siswa dan guru. Hasilnya menunjukkan bahwa sistem ini sangat layak digunakan, dengan nilai kelayakan keseluruhan mencapai 91,3%. Penerapan sistem ini memberikan kemudahan dalam manajemen data pendaftaran, pengumuman hasil, serta informasi umum sekolah secara *daring*, yang sebelumnya hanya mengandalkan formulir *Google* dengan fitur terbatas. Uji

validitas dan pengujian sistem menggunakan *metode black box* menunjukkan bahwa sistem berfungsi sesuai dengan yang dirancang, mendukung proses digitalisasi administrasi pendidikan di wilayah terpencil.

Penelitian ketiga oleh (Sulistio & Fitri, 2020) dengan judul "Rancang Bangun Sistem Informasi Penerimaan Peserta Didik Baru Berbasis *Web* Pada SDIT Al-Manar Kota Pekanbaru". Sistem sebelumnya masih dilakukan secara manual, yang menimbulkan berbagai kendala seperti tulisan tangan yang tidak terbaca, proses pendaftaran yang merepotkan, dan pengelolaan data yang kurang efisien. Dalam penelitian ini, penulis menggunakan metode observasi, wawancara, dan studi pustaka, kemudian membangun sistem menggunakan *CodeIgniter* dan *Bootstrap* dengan pengujian *black box* dan *white box*. Sistem yang dirancang memungkinkan proses pendaftaran, pencatatan pembayaran, dan pembagian kelas dilakukan secara terintegrasi, sehingga tidak hanya meningkatkan efisiensi administrasi, tetapi juga memudahkan orang tua dalam mengakses informasi secara *online*. Tampilan antarmuka mencakup halaman *login*, formulir pendaftaran, pencatatan pembayaran, pembagian kelas, dan laporan yang bisa dicetak. Hasil implementasi menunjukkan bahwa sistem ini memberikan kemudahan dan efektivitas yang lebih tinggi dalam pelaksanaan PPDB dibandingkan metode sebelumnya, serta mampu mendukung transformasi digital di lingkungan pendidikan dasar (Sulistio & Fitri, 2020).

Penelitian keempat yang dilakukan oleh (Almufqi dkk., 2023) merancang sistem informasi penerimaan peserta didik baru (PPDB) berbasis *web* di SMK Taruna Karya 1 Karawang untuk menggantikan proses pendaftaran manual yang selama ini belum efisien. Dengan menggunakan pendekatan *Waterfall* dalam pengembangan perangkat lunak, sistem ini dibangun melalui tahapan analisis kebutuhan, desain, pengkodean dengan *PHP, HTML, CSS,* dan *JavaScript*, serta pengujian melalui metode *black-box* dan *usability testing*. Sistem ini tidak hanya memfasilitasi proses pendaftaran *online* tetapi juga menyediakan fitur pelacakan status pendaftaran, pengelolaan data oleh admin, serta integrasi profil sekolah untuk memperluas informasi kepada masyarakat. Pengujian menunjukkan sistem ini valid secara fungsional dan memiliki tingkat kemudahan penggunaan yang tinggi bagi pengguna. Penelitian ini menegaskan pentingnya digitalisasi dalam manajemen administrasi pendidikan untuk meningkatkan efisiensi, transparansi, dan

keterjangkauan informasi pendidikan bagi masyarakat luas (Almufqi et al., 2023).

Penelitian kelima oleh (Meliana dkk., 2025) mengembangkan sistem informasi berbasis *web* untuk pendaftaran, tes masuk, dan penerimaan peserta didik baru di TK Mekar Sari Desa Ambawang guna mengatasi proses konvensional yang memakan waktu dan kurang efisien. Sistem ini dirancang menggunakan metode pengembangan *Waterfall* dengan tahapan analisis, desain, implementasi, pengujian menggunakan *black-box testing*, serta penerapan. Dibangun menggunakan *PHP native* dan *MySQL*, sistem ini menyediakan fitur lengkap mulai dari registrasi akun, *login*, pengisian formulir, penjadwalan tes, hingga laporan kelulusan yang dikelola oleh admin. Fitur tambahan seperti informasi profil sekolah, agenda kegiatan, dan struktur organisasi juga tersedia untuk publik pada tingkat pengguna umum (*free user*). Hasil pengujian menunjukkan sistem berjalan sesuai harapan dengan validitas *input-output* yang baik, serta mempermudah proses administrasi, mengurangi beban tenaga kerja, dan memperluas akses informasi kepada masyarakat. Penelitian ini menekankan pentingnya transformasi digital dalam pendidikan anak usia dini, sekaligus memberi model implementasi sistem informasi yang aplikatif dan terjangkau di tingkat pendidikan dasar (Meliana dkk., 2025).

Tabel 2. 1 Perbandingan variabel penelitian

Peneliti dan Tahun	Judul	Metode RPL	Metode Pengujian	Hasil	Fitur
(Khasbulloh & Karim, 2023)	Rancang Bangun Sistem Informasi Penerimaan Peserta Didik Baru Berbasis *Web* menggunakan Framework Laravel	*RAD (Rapid App Dev)*	*Black-box testing*	Sistem dinyatakan valid dan sesuai kebutuhan pengguna.	Pendaftaran online, upload dokumen, cetak formulir, pengumuman kelulusan, verifikasi admin.
Melati, Sahiruddin, & Ramadhani (2024)	Rancang Bangun Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) Pada SMK Muhammadiyah Salawati Berbasis	*Waterfall*	*Black-box testing*	Sistem sangat layak digunakan, skor kelayakan 91,3%.	Pendaftaran online, cetak bukti pendaftaran, dashboard admin, pengelolaan data pendaftar dan sekolah.

Peneliti dan Tahun	Judul	Metode RPL	Metode Pengujian	Hasil	Fitur
	Website Menggunakan Metode Waterfall				
Sulistio & Fitri (2020)	Rancang Bangun Sistem Informasi Penerimaan Peserta Didik Baru Berbasis Web Pada SDIT Al-Manar Kota Pekanbaru	*Custom SDLC*	*Black-box & White-box*	Sistem meningkatkan efisiensi dan mempermudah akses informasi.	Pendaftaran, pencatatan pembayaran, pembagian kelas, laporan siswa diterima, laporan keuangan.
Almufqi, Voutama, & Heryana (2023)	Rancang Bangun Sistem Penerimaan Peserta Didik Baru Berbasis *Web* Pada SMK Taruna Karya 1 Karawang	*Waterfall*	*Black-box & Usability testing*	*Website* valid dan *usability* tinggi; memudahkan calon siswa dan admin.	Pendaftaran *online*, status pendaftaran, profil sekolah, akses *WhatsApp*, admin dapat ekspor data, edit & hapus data.
Meliana, Muin, & Setiadi (2025)	Sistem Informasi Pendaftaran, Tes Siswa Masuk dan Penerimaan Peserta Didik baru Berbasis *Web* Pada TK Mekar Sari Desa Ambawang.	*Waterfall*	*Black-box testing*	Sistem valid, efisien dalam waktu dan tenaga, meningkatkan akses informasi.	*Free user* (informasi umum), *login*, registrasi akun, form pendaftaran, tes masuk, kelulusan, *dashboard* admin, laporan, CRUD data akun & guru, pembagian kelas A & B.
Arindi Frestisia Ningtias (2025)	Rancang Bangun Sistem Informasi Penerimaan Peserta Didik Baru Menggunakan	*Prototype*	*Black-box testing*	Harapan: Dengan adanya sistem ini, diharapkan pelayanan informasi menjadi lebih	Notifikasi Real-Time, Sistem Pemilihan Jadwal Wawancara, Dashboard Data

Peneliti dan Tahun	Judul	Metode RPL	Metode Pengujian	Hasil	Fitur
	Metode Prototyping (Studi Kasus Raudhatul Athfal Fadhilah)			cepat, akurat, dan modern, serta mampu meningkatkan efisiensi administrasi dan citra profesional lembaga di mata masyarakat.	Analitik untuk Sekolah.

Berdasarkan penelitian terdahulu pada Tabel 2.1 Perbandingan Variabel Penelitian, diketahui bahwa beberapa penelitian telah membahas sistem informasi PPDB berbasis *web*. Namun, masih terdapat keterbatasan pada fitur notifikasi *real-time*, pemilihan jadwal wawancara, *dashboard* analitik, serta *chatbot* untuk membantu pengguna memperoleh informasi PPDB dan profil sekolah secara cepat.

Penelitian ini berbeda karena mengembangkan sistem informasi profil sekolah dan PPDB berbasis *web* pada RA Fadhilah menggunakan metode *prototyping*. Sistem ini dilengkapi pendaftaran *online*, notifikasi *real-time*, pemilihan jadwal wawancara, *dashboard* analitik, dan *chatbot*.