

BAB II

LANDASAN TEORI

2.1 Posisi Penelitian

Tabel 2. 1 Posisi Penelitian (*State of the art*)

No	Peneliti (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Perbandingan yang dijadikan alasan tinjauan penelitian
1	(Lestari et al., 2025) Vol. 6, No. 1, April 2025, pp. 33-42, DOI: https://doi.org/10.55122/junsibi.v6i1.1534 , p-ISSN: 2774-3446 e-ISSN: 2774-3454	<i>Pengembangan Sistem Informasi Akademik pada SMP NEGERI 3 TANAH PUTIH menggunakan framework Laravel</i> https://www.ejournal-ibik57.ac.id/index.php/junsibi/article/Detail/1534	-	SMP Negeri 3 Tanah Putih	Sistem dibangun untuk mengatasi pengolahan data manual, namun berfokus pada akademik saja
2	(Nur Aeni Hidayah, 2026) Volume 8, Nomor 1, Februari 2026 : 108-123, E-ISSN:2685-2594	<i>Rancang Bangun Sistem Informasi Akademik Berbasis Website (Studi Kasus: SMA Islam Harapan Ibu)</i> https://jurnalftik.unisi.ac.id/index.php/jupel/article/Detail/4751	Waterfall	SMA Islam Harapan Ibu	Sistem mampu meningkatkan efisiensi pengelolaan data akademik dan mempermudah akses informasi, namun masih berfokus pada data akademik dan belum mengintegrasikan Administrasi secara menyeluruh
3	(Fatman et al., 2023)	<i>Sistem Informasi Akademik berbasis Website menggunakan metode</i>	Waterfall	TK Santa Eka Puhu	Sistem dapat membantu pengelolaan data akademik, namun masih terbatas pada aspek

No	Peneliti (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Perbandingan yang dijadikan alasan tinjauan penelitian
	Volume 6 Nomor 1, Juni 2023, e-ISSN : 2614-1574 p-ISSN : 2621-3249	<i>Waterfall (Studi Kasus: TK Santa Eka Puhu)</i> https://journal.ipm2kpe.or.id/index.php/INTECOM/article/Detail/6202			akademik dan belum mencakup pengelolaan Administrasi secara terintegrasi
4	(Sobari et al., 2026) Vol. 10 No.1, Februari 2026	<i>Rancang Bangun Sistem Informasi Akademik Siswa Berbasis Website di DTA Miftahul Falah</i> https://ejournal.iain.ac.id/jati/article/Detail/16886	<i>Prototype</i>	DTA Miftahul Falah	Sistem mampu mengatasi pengolahan data manual dan meningkatkan efisiensi, namun menggunakan metode <i>Prototype</i> sehingga kurang menekankan dokumentasi sistem secara terstruktur dibandingkan metode <i>Waterfall</i>
5	(Arafat & Pratama, 2025) Vol. 9, No. 4, September 2024 (173-178), DOI: https://doi.org/10.32493/informatika.v9i4.46008 , ISSN: 2541-1004 e-ISSN: 2622-4615	<i>Sistem Informasi Akademik Berbasis Website Menggunakan Metode Pengembangan Sistem Extreme Programming</i> https://openjournal.unpam.ac.id/index.php/informatika/article/Detail/46008	<i>Extreme Programming</i>	Madrasah Aliyah Andalusia	Sistem mampu meningkatkan efisiensi pengelolaan data akademik, namun menggunakan metode <i>Extreme Programming</i> yang lebih fleksibel sehingga kurang menekankan dokumentasi tahapan secara terstruktur dibandingkan metode <i>Waterfall</i>

No	Peneliti (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Perbandingan yang dijadikan alasan tinjauan penelitian
6	(Fadhilah et al., 2026) Vol. 6, No.1, Mei 2026, p- ISSN:2808-7550 e- ISSN:2798-0235	<i>Perancangan Sistem Informasi Akademik Berbasis Website pada PAUD Permata Hati Bojonegoro</i> https://journal.stiestekom.ac.id/index.php/mifortekh/article/Detail/1014	RAD	PAUD Permata Hati Bojonegoro	Sistem mampu meningkatkan efisiensi pengelolaan data akademik dan komunikasi, namun menggunakan metode RAD yang bersifat iteratif sehingga kurang menekankan dokumentasi tahapan secara sistematis dibandingkan metode <i>Waterfall</i>
7	(Pangri et al., 2025) Volume 6 No 4 Agustus 2025, Page 2357-2367, ISSN 2808-005X	<i>Perancangan Sistem Informasi Akademik TK Berbasis Website: Studi pada TK Ekklesia</i> https://ejournal.sisfokomtek.org/index.php/jumin/article/Detail/6647	<i>Waterfall</i>	TK Ekklesia	Sistem mampu meningkatkan efisiensi pengelolaan data dengan metode <i>Waterfall</i> yang terstruktur, namun tidak secara spesifik membahas sistem informasi akademik dan Administrasi secara terintegrasi
8	(Seta et al., 2021) Vol. 5, No.1, Oktober 2021 P-ISSN 2580-8737, E-ISSN 2623-64X	<i>Rancang Bangun Sistem Informasi Administrasi Keuangan Sekolah Berbasis Website Menggunakan Framework Laravel di SMPI</i>	SDLC / Website Based	Sistem Administrasi di SMPI Lukman Hakim PakiSaji	Sistem mampu meningkatkan efisiensi pengelolaan data Administrasi, namun belum secara spesifik mengintegrasikan sistem akademik dan Administrasi dalam satu platform terpadu

No	Peneliti (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Perbandingan yang dijadikan alasan tinjauan penelitian
		<i>LukmanHakim PakiSaji</i> https://ejournal.uniramalang.ac.id/g-tech/article/Detail/1152			
9	(Fernandy et al., 2022) Vol 1 No.1 (2022) – PISSN: 2808-9375 EISSN: 2808-8999	<i>Rancang Bangun Sistem Informasi Websitesite Program Studi Teknik Informatika UNUSIA Menggunakan Metode Waterfall dan Framework Laravel</i> https://journalcenter.org/index.php/jupikom/article/Detail/230	<i>Waterfall</i>	Program Studi Teknik Informatika UNUSIA	Sistem mampu menyediakan informasi secara dinamis menggunakan <i>Laravel</i> dan <i>Waterfall</i> , namun lebih berfokus pada media informasi program studi dan belum mengintegrasikan pengelolaan data akademik dan Administrasi secara lengkap
10	Neneng Rizqoh (2026)	<i>Rancang Bangun Sistem Informasi Akademik Berbasis Website Menggunakan Metode Waterfall di Ruang Belajar Sa'idah</i>	<i>Waterfall</i>	Ruang Belajar Sa'idah	Penelitian ini mengintegrasikan pengelolaan data akademik dalam satu sistem berbasis <i>Website</i> yang terpusat, menggunakan <i>framework Laravel</i> dengan metode <i>Waterfall</i> yang terstruktur dan terdokumentasi, serta disesuaikan dengan kebutuhan nyata pada lembaga

No	Peneliti (Tahun)	Judul Penelitian	Metode	Objek Penelitian	Perbandingan yang dijadikan alasan tinjauan penelitian
					pendidikan non- formal

Berdasarkan hasil analisis terhadap penelitian-penelitian terdahulu, diketahui bahwa sebagian besar penelitian berfokus pada pengembangan sistem informasi akademik berbasis *website* sebagai solusi untuk mengatasi proses pengelolaan data yang masih dilakukan secara manual. Sistem yang dikembangkan pada umumnya telah mampu meningkatkan efisiensi serta ketepatan dalam pengelolaan data akademik, seperti data siswa, data pengajar, nilai, dan jadwal. Meskipun demikian, beberapa penelitian masih memiliki keterbatasan karena belum mengintegrasikan aspek administrasi secara menyeluruh serta belum sepenuhnya disesuaikan dengan karakteristik dan kebutuhan masing-masing objek penelitian.

Selain itu, beberapa penelitian menerapkan metode pengembangan seperti *Prototype*, *Extreme Programming*, dan RAD yang menawarkan fleksibilitas dalam proses pengembangan. Namun, metode-metode tersebut cenderung kurang menekankan dokumentasi setiap tahapan secara sistematis. Disisi lain, penelitian yang menggunakan *framework Laravel* dengan metode *Waterfall* umumnya masih berorientasi pada penyampaian informasi atau pengelolaan data akademik, tanpa mengintegrasikan seluruh kebutuhan sistem ke dalam satu platform yang terpadu.

Berdasarkan kondisi tersebut, kebaruan (*novelty*) dalam penelitian ini terletak pada pengembangan sistem informasi Ruang Belajar Sa'idah berbasis *website* yang mampu mengintegrasikan seluruh proses pengelolaan data ke dalam satu sistem terpusat. Sistem yang dikembangkan mencakup pengelolaan data siswa, data pengajar, mata pelajaran, jadwal dan pengumuman, absensi, nilai, materi harian, administrasi, hingga *e-raport*. Selain itu, sistem juga dilengkapi dengan fitur pengiriman notifikasi email otomatis kepada orang tua sebagai media penyampaian informasi akademik, seperti konfirmasi pembayaran infaq, materi harian, rekap harian yang memuat data absensi dan infaq, serta penerbitan *e-raport*. Proses pengembangan sistem dilakukan menggunakan *framework Laravel* dengan metode *Waterfall* sehingga setiap tahapan dapat dilaksanakan secara terstruktur, terdokumentasi, dan disesuaikan dengan kebutuhan nyata pada lembaga pendidikan nonformal, yaitu Ruang Belajar Sa'idah. Dengan demikian, sistem yang dihasilkan diharapkan mampu memberikan solusi yang lebih efektif, efisien, serta sesuai dengan kebutuhan operasional Ruang Belajar Sa'idah dibandingkan proses pengelolaan yang digunakan sebelumnya.

2.2 Penelitian Terdahulu

Berbagai penelitian mengenai sistem informasi akademik berbasis *website* telah dilakukan oleh peneliti sebelumnya dengan menerapkan beragam metode dan pendekatan pengembangan. Penelitian (Lestari et al., 2025) menunjukkan bahwa sistem informasi akademik yang dikembangkan mampu mengurangi kendala dalam proses pengelolaan data yang sebelumnya masih dilakukan secara manual. Namun sistem tersebut masih difokuskan pada pengelolaan data akademik dan belum mencakup aspek lainnya. Selanjutnya, penelitian oleh (Nur Aeni Hidayah, 2026) menunjukkan bahwa penggunaan metode *Waterfall* dalam pengembangan sistem informasi mampu meningkatkan efisiensi proses pengelolaan data serta memberikan kemudahan bagi pengguna dalam mengakses informasi.

Penelitian lain dilakukan oleh (Fatman et al., 2023) menunjukkan bahwa sistem informasi akademik berbasis *website* dapat membantu mengatasi pencatatan manual, namun fitur yang dihasilkan masih tergolong sederhana. Sementara itu, penelitian oleh (Sobari et al., 2026) menggunakan metode *Prototype* yang mampu meningkatkan efisiensi pengolahan data, tetapi kurang menekankan dokumentasi sistem secara terstruktur dibandingkan metode *Waterfall*. Penelitian oleh (Arafat & Pratama, 2025) menggunakan metode *Extreme Programming* yang bersifat fleksibel, namun memiliki kelemahan dalam dokumentasi tahapan pengembangan yang kurang sistematis.

Selain itu, penelitian (Fadhilah et al., 2026) menggunakan metode RAD yang mampu mempercepat proses pengembangan sistem, namun kurang menekankan pada tahapan yang terstruktur. Penelitian oleh (Pangri et al., 2025) menunjukkan bahwa metode *Waterfall* mampu menghasilkan sistem yang terstruktur, namun belum sepenuhnya disesuaikan dengan kebutuhan pengguna secara spesifik. Disisi lain, penelitian oleh (Seta et al., 2021) lebih berfokus pada sistem Administrasi keuangan, sehingga belum mencakup sistem akademik secara menyeluruh. Sedangkan penelitian oleh (Fernandy et al., 2022) menunjukkan bahwa sistem berbasis *website* mampu menyediakan informasi secara dinamis, namun masih berfokus pada media informasi dan belum mengelola data akademik secara lengkap.

Berdasarkan beberapa penelitian tersebut, dapat disimpulkan bahwa sistem informasi akademik berbasis *website* mampu meningkatkan efisiensi dan efektivitas pengelolaan data. Namun, masih terdapat beberapa keterbatasan, seperti belum terintegrasinya seluruh kebutuhan sistem, kurangnya penyesuaian dengan kebutuhan pengguna, serta belum optimalnya dokumentasi pengembangan sistem pada setiap tahapan. Oleh karena itu, penelitian ini dilakukan untuk mengembangkan sistem informasi akademik berbasis *website* menggunakan

metode *Waterfall* yang lebih terstruktur dan disesuaikan dengan kebutuhan nyata pada objek penelitian.

2.3 Kajian Teori

Kajian teori merupakan landasan konseptual yang digunakan dalam penelitian ini sebagai dasar dalam memahami permasalahan serta mendukung proses perancangan sistem yang akan dikembangkan. Pada bagian ini akan dijelaskan berbagai konsep dan teori yang relevan, meliputi rancang bangun sistem, sistem informasi, *website*, basis data, teknologi pengembangan *website*, *framework Laravel*, serta metode pengembangan perangkat lunak yang digunakan. Dengan adanya kajian teori ini, diharapkan penelitian yang dilakukan memiliki dasar ilmiah yang jelas dan terarah sesuai dengan tujuan penelitian.

2.4 Rancang Bangun Sistem

Rancang bangun adalah proses merancang dan mengembangkan suatu sistem secara sistematis, dimulai dari analisis kebutuhan, perancangan, hingga tahap implementasi. Proses tersebut bertujuan menghasilkan sistem yang sesuai dengan kebutuhan pengguna serta mampu memberikan solusi terhadap permasalahan yang dihadapi (Seta et al., 2021).

2.5 Sistem Informasi

2.5.1 Pengertian Sistem Informasi

Menurut Leitch Roses dalam penulis buku (Angelilla et al., 2023) Sistem informasi merupakan suatu sistem dalam organisasi yang berfungsi untuk mengelola transaksi harian, mendukung kegiatan operasional dan administratif sesuai dengan karakteristik organisasi, serta membantu dalam penyusunan dan penyampaian laporan yang dibutuhkan.

2.5.2 Komponen Sistem Informasi

Sistem informasi terdiri atas beberapa komponen yang saling berhubungan dan bekerja secara terintegrasi untuk menghasilkan informasi yang bermanfaat. Adapun komponen-komponen sistem informasi meliputi sebagai berikut (Syafrial Fachri Pane, Mochamad Zamzam, 2020).

- a. Komponen *Input*, yaitu data atau informasi yang dimasukkan ke dalam sistem untuk selanjutnya diproses.
- b. Komponen *Model*, yaitu kumpulan prosedur, aturan, logika, dan metode yang digunakan untuk mengolah data menjadi informasi sesuai dengan tujuan yang telah ditentukan.
- c. Komponen *Output*, yaitu hasil pengolahan data berupa informasi yang dapat dimanfaatkan oleh pengguna dalam mendukung proses pengambilan keputusan.
- d. Komponen teknologi, yaitu perangkat yang digunakan untuk mendukung operasional sistem, meliputi *hardware*, *software*, dan jaringan yang berfungsi untuk menerima, mengolah, menyimpan, serta mendistribusikan informasi.

- e. Komponen basis data, yaitu sekumpulan data yang saling berhubungan dan tersimpan secara terstruktur menggunakan sistem manajemen basis data sehingga dapat diakses dan dikelola dengan mudah.
- f. Komponen kontrol, yaitu mekanisme pengendalian yang berfungsi untuk menjaga keamanan, keakuratan, dan kelancaran proses dalam sistem informasi dari berbagai gangguan atau kesalahan.

2.6 Sistem Informasi Akademik

2.6.1 Pengertian Sistem Informasi Akademik

Sistem informasi akademik adalah sistem yang dirancang untuk mendukung pengelolaan administrasi serta penyediaan informasi yang berkaitan dengan kegiatan akademik. Melalui sistem ini, proses administrasi dapat dilakukan secara lebih terstruktur sehingga informasi dapat diakses dengan lebih cepat, tepat, dan efisien (Yusuf Ali et al., 2024)

2.7 Website

2.7.1 Pengertian Website

Menurut Lukman Hakim dalam penulis buku (Kusumawardani et al., 2023) *Website* merupakan salah satu layanan yang tersedia pada internet yang berfungsi sebagai media untuk menyajikan dan menghubungkan berbagai informasi dalam bentuk dokumen digital. Setiap dokumen atau halaman yang terdapat pada *website* disebut *webpage*, sedangkan link atau tautan digunakan sebagai penghubung agar pengguna dapat berpindah dari satu halaman ke halaman lainnya, baik dalam satu *server* maupun pada *server* yang berbeda. Untuk mengakses halaman *website*, pengguna dapat memanfaatkan aplikasi peramban (*browser*) seperti Google Chrome, Mozilla Firefox, dan *browser* lainnya.

2.7.2 Jenis-jenis Website

Menurut (Abdulloh, 2022), *website* dibedakan menjadi beberapa jenis berdasarkan cara pengelolaan informasi dan tingkat interaksi yang diberikan kepada pengguna, yaitu *website* statis, *website* dinamis, dan *website* interaktif.

a. Website Statis

Website statis merupakan *website* yang isi atau kontennya relatif tetap dan tidak mengalami perubahan secara berkala. Proses pengelolaan konten pada *website* statis umumnya dilakukan langsung oleh pengembang atau pemilik *website* melalui perubahan pada kode sumber. *Website* statis umumnya dimanfaatkan untuk menyajikan informasi yang bersifat sederhana dan jarang mengalami perubahan. Jenis *website* ini banyak digunakan sebagai media penyampaian profil perusahaan, profil organisasi, maupun halaman *Landing Page*.

b. Website Dinamis

Website dinamis adalah jenis *website* yang kontennya dapat diperbarui secara berkala oleh pengelola atau pemilik *website* melalui sistem pengelolaan tertentu. *website* ini memungkinkan informasi yang ditampilkan berubah sesuai dengan kebutuhan pengguna dan

perkembangan data. *Website* dinamis banyak digunakan oleh perusahaan maupun perorangan yang membutuhkan pembaruan informasi secara rutin, seperti blog, portal berita, dan sistem informasi berbasis *website*.

c. *Website* Interaktif

Website interaktif adalah jenis *website* yang dirancang untuk mendukung komunikasi dua arah antara pengguna dan sistem. Melalui *website* ini, pengguna tidak hanya menerima informasi, tetapi juga dapat berinteraksi dengan sistem, memberikan masukan, serta mengelola data tertentu sesuai dengan hak akses yang dimilikinya. Contoh *website* interaktif antara lain media sosial, forum diskusi, serta platform *e-commerce* dan *marketplace*.

2.8 Teknologi Pengembangan Sistem

2.8.1 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) adalah satu bahasa pemrograman yang banyak dimanfaatkan dalam pengembangan aplikasi berbasis *website*. Bahasa pemrograman PHP membantu pengembang untuk membuat halaman *website* yang dinamis dan interaktif melalui pengolahan data di sisi *server*. Bahasa pemrograman PHP juga memiliki *syntaks* yang relatif mudah dipahami, mendukung integrasi dengan berbagai sistem manajemen basis data, serta dapat dijalankan pada berbagai sistem operasi (Nafisah & Ujianti, 2025).

2.8.2 HTML (*Hypertext Markup Language*)

HTML (*Hypertext Markup Language*) adalah bahasa markup yang digunakan untuk membangun serta menampilkan struktur dasar pada sebuah halaman *website*. HTML menyediakan berbagai elemen yang memungkinkan pengembang untuk menampilkan teks, gambar, tabel, tautan, audio, video, dan komponen multimedia lainnya. Dalam pengembangan aplikasi *website*, HTML umumnya dikombinasikan dengan bahasa pemrograman dan teknologi lain untuk mengatur logika aplikasi dan pengelolaan data (Nafisah & Ujianti, 2025).

2.8.3 CSS (*Cascading Style Sheets*)

CSS (*Cascading Style Sheets*) adalah bahasa *stylesheet* yang berperan dalam mengatur elemen visual pada halaman *website*, seperti tata letak, warna, ukuran, serta tampilan komponen agar halaman web menjadi lebih rapi dan mudah digunakan. Dalam pengembangan *website*, CSS umumnya digunakan bersama HTML atau XHTML untuk memisahkan struktur konten dan aspek tampilannya sehingga halaman web menjadi lebih terorganisasi dan mudah dikelola (Koesheriyatin, 2014).

2.9 Framework Laravel

2.9.1 Pengertian *Laravel*

Laravel merupakan salah satu *Framework* berbasis PHP yang bersifat *open source* dan dapat digunakan secara bebas oleh pengembang dalam membangun aplikasi berbasis *website*. *Framework* ini pertama kali dikembangkan oleh Taylor Otwell dan diperkenalkan pada tahun 2011. Seiring dengan perkembangan teknologi serta dukungan komunitas luas, *Laravel* mengalami peningkatan popularitas

dan banyak dimanfaatkan dalam proses pengembangan perangkat lunak. Hingga saat ini, laravel menjadi salah satu *framework* PHP yang banyak digunakan karena memiliki berbagai fitur pendukung serta mampu bersaing dengan *framework* lainnya (Abdulloh, 2025).

2.9.2 Keunggulan *Laravel*

Menurut (Abdulloh, 2022), *Laravel* memiliki berbagai keunggulan yang menjadikannya salah satu *framework* PHP yang banyak digunakan dalam pengembangan aplikasi web, diantaranya:

- a. *Laravel* menyediakan berbagai fitur yang tidak dimiliki oleh *framework* lain, seperti *artisan*, *migration*, *blade*, dan beragam fitur pendukung lainnya.
- b. *Laravel* merupakan *framework* PHP yang memiliki sintaks yang sederhana dan mudah dipahami. Hal ini memudahkan pengembang, terutama pemula, untuk mempelajari dan menggunakannya.
- c. *Laravel* didukung oleh dokumentasi yang lengkap dan terstruktur. Setiap versi *Laravel* memiliki dokumentasi tersendiri yang mencakup panduan instalasi hingga penggunaan berbagai fitur yang tersedia.
- d. *Laravel* yang digunakan oleh banyak pengembang di seluruh dunia sehingga tersedia banyak *Library* dan dukungan komunitas yang terus berkembang.
- e. *Laravel* terintegrasi dengan *Composer*, yaitu alat manajemen dependensi PHP yang memudahkan pengembang dalam mengunduh, mengelola, dan memasang berbagai *Library* yang dibutuhkan.
- f. *Laravel* memiliki *Template engine* bernama *Blade* yang membantu pengembang dalam mengelola dan menampilkan data pada halaman HTML secara lebih mudah dan efisien.

2.9.3 Arsitektur MVC dalam *Laravel*

Laravel menerapkan pola arsitektur *Model-View-Controller* (MVC) yang membagi logika aplikasi ke dalam tiga komponen utama, yaitu *Model*, *View*, dan *Controller* (Kurniawan, n.d.).

- a. *Model* berfungsi untuk mengelola data, logika bisnis, dan aturan yang diterapkan dalam aplikasi. Komponen ini berinteraksi dengan basis data serta memproses data yang akan dikirimkan atau diterima oleh *Controller*.
- b. *View* merupakan komponen yang bertugas menampilkan antarmuka atau presentasi data kepada pengguna. Data yang ditampilkan berasal dari *Controller* agar dapat disajikan dalam bentuk yang mudah dipahami oleh pengguna.
- c. *Controller* berperan sebagai penghubung antara *Model* dan *View*. Komponen ini menerima permintaan dari pengguna, mengolahnya dengan bantuan *Model*, kemudian meneruskan hasilnya ke *Detail* untuk ditampilkan.

2.10 AdminLTE

AdminLTE merupakan template antarmuka administratif (admin *Dashboard* template) yang dibangun menggunakan *framework Bootstrap*. AdminLTE menyediakan berbagai

komponen antarmuka yang siap digunakan, seperti *Dashboard*, *sidebar*, formulir, tabel, *widget*, sehingga memudahkan pengembang dalam membangun aplikasi *web* dengan tampilan yang responsif dan profesional. Penggunaan AdminLTE memungkinkan proses pengembangan menjadi lebih efisien karena pengembang dapat berfokus pada implementasi fungsionalitas sistem tanpa harus merancang antarmuka dari awal (Audrey et al., 2024).

2.11 Database

2.11.1 Pengertian Database

Menurut C. J. Date, *Database* merupakan kumpulan data yang tersimpan secara terkomputerisasi. Sistem *Database* pada dasarnya bertujuan untuk mengelola serta memelihara data tersebut agar tetap terorganisir dan dapat diakses dengan mudah ketika diperlukan (Yani & Saputra, 2018).

2.11.2 MySQL

MySQL merupakan sistem manajemen basis data yang digunakan untuk menyimpan, mengelola, dan mengolah data secara terstruktur. *MySQL* mendukung berbagai operasi basis data, seperti penambahan, pengubahan, pencarian, dan penghapusan data, serta kompatibel dengan berbagai sistem operasi dan bahasa pemrograman (Nafisah & Ujianti, 2025).

2.12 SMTP (Simple Mail Transfer Protocol)

Simple Mail Transfer Protocol (SMTP) adalah protokol standar yang digunakan untuk mengirimkan email melalui jaringan internet. SMTP berfungsi meneruskan pesan dari pengirim ke *server* email tujuan dengan melakukan verifikasi terhadap alamat email penerima sebelum proses pengiriman dilakukan. Jika alamat penerima valid, maka email akan dikirimkan ke tujuan (Rubedo et al., 2020).

2.13 DomPDF

2.13.1 Pengertian DomPDF

Dompdf adalah *Library* PHP yang digunakan untuk menghasilkan dokumen PDF dari tampilan HTML dan CSS. Kemudahan dalam proses integrasinya menjadikan *Dompdf* sebagai salah satu solusi yang banyak digunakan untuk kebutuhan pembuatan laporan pada aplikasi berbasis *website* (Muthohir, 2025).

2.14 Tools Pendukung Pengembangan Website

2.14.1 Draw IO

Draw.io merupakan platform berbasis *website* yang digunakan untuk membuat berbagai jenis diagram secara online. Aplikasi ini menyediakan fitur yang cukup lengkap untuk mendukung proses perancangan sistem yang dilakukan oleh pengembang. Draw.io dapat diakses melalui browser sehingga memudahkan pengguna dalam penggunaannya.

Dalam penggunaannya, Draw.io dapat digunakan untuk membuat berbagai diagram UML, seperti *Use case diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*. Selain itu, Draw.io

juga dapat terintegrasi dengan *Google Drive*, sehingga setiap diagram yang dibuat dapat langsung di simpan dalam satu *file* di penyimpanan tersebut.

Draw.io juga menyediakan berbagai pilihan format ekspor, seperti PNG, JPG, dan XML. Untuk keperluan pengeditan kembali, disarankan menyimpan *file* dalam format XML agar dapat dibuka dan dimodifikasi di kemudian hari (Halifah & Subrata, 2025).

2.14.2 XAMPP

Menurut Jogiyanto (2005, dalam (Joni, 2019)), *XAMPP* adalah perangkat lunak berbasis open source yang digunakan sebagai perangkat pendukung beberapa komponen penting seperti *web server*, PHP, dan MySQL yang sudah terintegrasi dalam satu paket, sehingga pengguna tidak perlu melakukan proses instalasi maupun pengaturan setiap komponen secara terpisah. Dengan adanya *XAMPP*, proses pembuatan dan pengujian *website* pada lingkungan lokal dapat dilakukan dengan lebih mudah.

2.14.3 Visual Studio Code

Visual Studio Code adalah aplikasi penyunting kode yang dikembangkan oleh Microsoft dan dapat dijalankan pada berbagai platform sistem operasi, seperti Windows, Linux, maupun macOS. Aplikasi ini telah dilengkapi dengan dukungan bawaan untuk JavaScript, TypeScript, dan Node.js, serta menyediakan berbagai ekstensi yang memungkinkan pengembangan menggunakan bahasa pemrograman lain, seperti C++, C#, Python, dan PHP. Dengan desain yang ringan namun kaya fitur, Visual Studio Code banyak digunakan oleh pengembang untuk membangun dan mengembangkan berbagai jenis aplikasi, terutama aplikasi berbasis *web* (Pasha, 2023).

2.15 Metode Waterfall

2.15.1 Pengertian Waterfall

Waterfall adalah suatu pendekatan dalam pengembangan sistem informasi. Proses ini mensyaratkan bahwa pengerjaannya dilakukan dalam urutan yang teratur atau sekuensial, dimulai dari tahap analisis kebutuhan, perancangan sistem, pengembangan sistem, pengujian sistem dan pemeliharaan sistem (Ramadhan et al., 2023).

2.15.2 Tahapan Metode Waterfall

Adapun tahapan tahapan metode *Waterfall* sebagai berikut (Ramadhan et al., 2023).

a. Analisis Kebutuhan

Tahap ini dilakukan proses pengumpulan dan analisis informasi untuk mengetahui kebutuhan sistem, baik fungsional maupun nonfungsional yang diperlukan dalam pengembangan sistem.

b. Perancangan Sistem

Pada tahap ini, hasil analisis kebutuhan diterjemahkan ke dalam bentuk rancangan sistem yang meliputi desain antarmuka, struktur basis data, alur proses sistem sebagai acuan pada tahap implementasi.

c. Implementasi Sistem

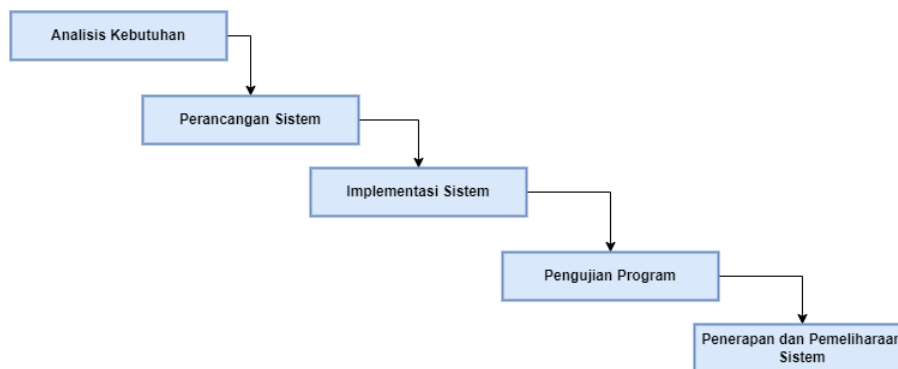
Tahap implementasi merupakan proses penerapan hasil perancangan ke dalam bentuk kode program sehingga menghasilkan aplikasi atau *website* yang dapat digunakan sesuai dengan fungsi yang telah dirancang.

d. Pengujian Sistem

Tahap pengujian dilakukan terhadap sistem yang telah dikembangkan untuk memastikan setiap fitur berjalan dengan baik serta mengetahui apakah sistem telah memenuhi kebutuhan dan kriteria yang telah ditetapkan.

e. Pemeliharaan Sistem

Tahap ini dilakukan apabila sistem mengalami kerusakan.



Gambar 2. 1 Tahapan Metode *Waterfall*

Sumber : Didaptasi dari (Muhammad Farhan , Ika Nur Fajri, 2026)

2.15.3 Kelebihan dan Kekurangan Metode *Waterfall*

Menurut (Afzil Ramadian , Safuan, 2025), metode *Waterfall* memiliki beberapa kelebihan dan kekurangan diantaranya:

Kelebihan metode *Waterfall*:

- Metode *Waterfall* mampu menghasilkan perangkat lunak dengan kualitas yang baik karena setiap tahapan dikerjakan secara terstruktur.
- Proses pengembangan dilakukan secara berurutan dan sistematis, sehingga dapat mengurangi kemungkinan terjadinya kesalahan.
- Dokumentasi selama proses pengembangan tersusun dengan rapi dan terorganisasi dengan baik.

Kekurangan metode *Waterfall*:

- Metode *Waterfall* memerlukan waktu dan biaya yang relatif besar selama proses pengembangan.
- Pelaksanaannya membutuhkan manajemen proyek yang baik agar setiap tahapan dapat berjalan sesuai rencana.
- Kesalahan yang tidak terdeteksi pada tahap awal dapat berdampak pada tahapan berikutnya dan menyebabkan perbaikan menjadi lebih sulit.
- Metode ini kurang sesuai untuk pengembangan produk yang membutuhkan proses yang cepat atau perubahan yang sering terjadi.

2.16 Unified Modelling Language (UML)

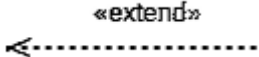
Unified Modelling Language (UML) merupakan suatu bahasa pemodelan standar yang digunakan dalam proses analisis, perancangan, serta dokumentasi pada pengembangan perangkat lunak. UML menyediakan berbagai jenis diagram yang dapat digunakan untuk mempresentasikan bagian atau aspek tertentu dari sebuah sistem, seperti *use case* diagram, *Activity* diagram, *Sequence* diagram, *class* diagram, dan *entity relationship* diagram (ERD). Dengan UML, pengembang dapat menggambarkan struktur, perilaku, serta interaksi yang terjadi di dalam sistem sehingga proses pengembangan menjadi terorganisasi dan mudah dipahami (Soni, 2024).

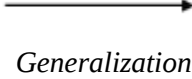
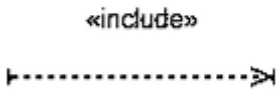
2.16.1 *Use case* diagram

Use case diagram merupakan salah satu jenis diagram pada UML yang digunakan untuk menggambarkan hubungan interaksi antara aktor sebagai pengguna dengan sistem yang dikembangkan. Diagram ini berfungsi untuk menjelaskan kebutuhan fungsional sistem dengan menampilkan fitur atau layanan yang tersedia keterlibatan pengguna dalam setiap fungsi tersebut (Soni, 2024).

Berikut simbol-simbol dari *Use case* diagram:

Tabel 2. 2 Simbol-Simbol *Use case* diagram

No	Simbol	Keterangan
1	 <i>Use case</i>	<i>Use case</i> digunakan untuk mempresentasikan fungsi atau layanan yang disediakan sistem guna memenuhi kebutuhan pengguna
2	 Aktor	Aktor merupakan entitas eksternal yang berinteraksi dengan sistem, baik berupa pengguna, proses, maupun sistem lain
3	 Asosiasi	Asosiasi menunjukkan hubungan interaksi antara aktor dan <i>Use case</i> yang menggambarkan keterlibatan aktor dalam suatu proses sistem
4	 «extend»	<i>Extend</i> menunjukkan hubungan perluasan, yaitu adanya fungsi tambahan yang dapat dijalankan pada

No	Simbol	Keterangan
		kondisi tertentu tanpa mengubah fungsi utama <i>Use case</i>
5		<i>Generalization</i> menunjukkan hubungan pewarisan antara elemen yang bersifat umum dengan elemen yang lebih spesifik
6		<i>Include</i> merupakan hubungan tambahan pada sebuah <i>Use case</i> , dimana suatu <i>Use case</i> utama membutuhkan <i>Use case</i> lain agar fungsi tertentu dapat dijalankan atau sebagai bagian yang harus dipenuhi dalam proses pelaksanaannya

Sumber: (Nanda Jarti, Alvendo Wahyu Aranski, 2026)

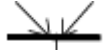
2.16.2 Activity Diagram

Activity Diagram merupakan salah satu jenis diagram dalam UML yang berfungsi untuk menggambarkan rangkaian aktivitas, alur kerja, serta proses pengambilan keputusan yang berlangsung dalam suatu sistem. Diagram ini membantu dalam memahami proses atau tahapan kerja sistem secara lebih jelas dan terstruktur (Soni, 2024).

Berikut simbol-simbol *Activity Diagram*:

Tabel 2. 3 Simbol-Simbol *Activity Diagram*

No	Simbol	Keterangan
1		Menunjukkan awal dari suatu aktivitas atau proses yang akan dijalankan dalam <i>Activity Diagram</i> . Umumnya simbol ini diletakkan pada bagian awal diagram.
2		Menandai berakhirnya seluruh rangkaian aktivitas atau proses dalam <i>Activity diagram</i> .

No	Simbol	Keterangan
3	 activities	Menggambarkan aktivitas, proses atau pekerjaan yang dilakukan oleh aktor maupun sistem selama proses berlangsung.
4	 Fork	Simbol yang digunakan untuk membagi satu alur proses menjadi beberapa aktivitas yang dapat dijalankan secara bersamaan (pararel)
5	 Join	Simbol yang menggabungkan kembali beberapa alur aktivitas pararel menjadi satu alur proses sebelum melanjutkan ke aktivitas berikutnya.
6	 Decision	Menunjukkan percabangan proses berdasarkan suatu kondisi atau keputusan tertentu
7	 Swimlane	Membagi aktivitas berdasarkan aktor, pengguna, atau bagian yang bertanggung jawab, sehingga memudahkan dalam mengetahui peran masing-masing pada setiap proses berlangsung. VI

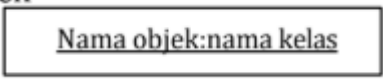
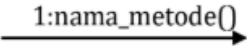
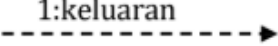
Sumber: (Alvian, Fadnan Ginanjar Yogi, Khusnuliawati, Hardika, Charolina, 2017)

2.16.3 Sequence Diagram

Sequence Diagram merupakan salah satu diagram dalam UML yang digunakan untuk memodelkan proses interaksi antarobjek dalam sistem berdasarkan alur waktu tertentu. Diagram ini menggambarkan komunikasi antarobjek melalui pertukaran pesan (message) yang terjadi pada suatu skenario proses (Soni, 2024).

Berikut simbol-simbol *Sequence Diagram*:

Tabel 2. 4 Simbol-Simbol *Sequence Diagram*

No	Simbol	Keterangan
1	 Aktor	Aktor merupakan entitas eksternal yang berinteraksi dengan sistem, baik berupa pengguna, proses, maupun sistem lain yang berada di luar batas sistem yang dikembangkan
2	 Lifeline	<i>Lifeline</i> digunakan untuk menunjukkan keberadaan suatu objek selama proses interaksi berlangsung dalam <i>Sequence Diagram</i>
3	 Objek	Objek adalah komponen yang terlibat dalam pertukaran pesan dan menjalankan fungsi tertentu dalam sistem
4	 Waktu aktif	Waktu aktif menunjukkan periode ketika objek sedang menjalankan suatu operasi atau memproses pesan yang diterima dari objek lain
5	Pesan tipe <i>call</i> 	Menunjukkan pemanggilan metode atau operasi yang dimiliki suatu objek
6	Pesan tipe <i>send</i> 	Menunjukkan pengiriman data atau informasi dari satu objek ke objek lainnya
7	Pesan tipe <i>return</i> 	Menunjukkan hasil atau nilai balikan dari proses yang telah dijalankan oleh suatu objek

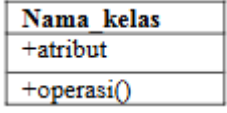
Sumber: (Nanda Jarti, Alvendo Wahyu Aranski, 2026)

2.16.4 Class diagram

Class diagram merupakan satu diagram UML yang digunakan untuk memvisualisasikan struktur statis dari sebuah sistem. Diagram ini menggambarkan elemen-elemen berupa kelas (*class*), atribut, metode, atau operasi yang dimiliki kelas, serta hubungan atau keterkaitan antar kelas dalam sistem (Soni, 2024).

Berikut simbol-simbol *Class Diagram*:

Tabel 2. 5 Simbol-Simbol *Class Diagram*

No	Simbol	Keterangan
1	 Kelas (<i>Class</i>)	Struktur yang menggambarkan objek dalam sistem beserta atribut dan metode yang dimilikinya
2	 Antarmuka (<i>interface</i>)	Penghubung yang mendefinisikan fungsi atau perilaku yang harus dimiliki oleh kelas
3	 Asosiasi (<i>Association</i>)	Hubungan antar kelas yang menunjukkan keterkaitan satu sama lain
4	 Asosiasi Berarah (<i>Directed Association</i>)	Hubungan antar kelas yang memiliki arah akses tertentu
5	 Generalisasi (<i>Generalization</i>)	Hubungan pewarisan antara kelas umum dan kelas khusus
6	 Ketergantungan (<i>Depedency</i>)	Hubungan dimana satu kelas bergantung pada kelas lainnya
7	 Agregasi (<i>Aggregation</i>)	Hubungan antara objek utama dengan objek yang menjadi bagian penyusunnya

Sumber: Diadaptasi dari (Aprianti & Maliha, 2016)

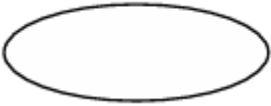
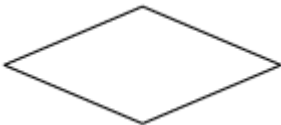
2.16.5 *Entity Relationship Diagram* (ERD)

Menurut Amalia, Sagita, dan Faisal (2021, dalam (Ernawati, 2023)) , ERD merupakan diagram yang digunakan untuk menggambarkan entitas-entitas yang terdapat dalam suatu sistem beserta hubungan (relasi) yang terjadi diantara entitas tersebut. ERD menyajikan model yang terdiri atas

kumpulan entitas, relasi, atribut yang mempresentasikan fakta-fakta dari kondisi nyata (*real world*) secara terstruktur dan sistematis sehingga memudahkan proses perancangan basis data.

Berikut simbol-simbol *Entity Relationship Diagram*:

Tabel 2. 6 Simbol-Simbol *Entity Relationship Diagram*

No	Simbol	Keterangan
1	 Entitas	Entitas merupakan sekumpulan objek atau data yang memiliki karakteristik tertentu dan dapat dibedakan atau diidentifikasi secara jelas dari objek lainnya
2	 Atribut	Atribut adalah bagian yang menjelaskan karakteristik atau informasi yang dimiliki oleh suatu entitas maupun relasi sebagai Detail pendukung
3	 Relasi	Relasi menggambarkan hubungan atau keterkaitan antara satu entitas dengan entitas lainnya dalam suatu sistem
4	 Garis	Garis berfungsi sebagai penghubung yang menunjukkan keterkaitan antara entitas, atribut, maupun relasi dalam sebuah diagram ERD

Sumber: (Ernawati, 2023)

2.17 Metode Pengujian Sistem

2.17.1 *Black Box Testing*

Black Box Testing adalah metode pengujian perangkat lunak yang dilakukan dengan melihat sistem dari sisi fungsionalitasnya tanpa perlu mengetahui detail proses internal maupun kode program yang digunakan. Dalam metode ini, pengujian dilakukan dengan memberikan masukan (*input*) tertentu kemudian mengamati hasil keluaran (*output*) yang dihasilkan oleh sistem (Durga Praveen Deevi, Naga Sushma Allur, Koteswararao Dondapati , Himabindu Chetlapalli, Thinagaran, 2024).

2.17.2 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) adalah tahapan pengujian terakhir yang dilakukan untuk mengevaluasi kesesuaian fungsi sistem berdasarkan kebutuhan pengguna. Pada tahap ini, sistem akan diuji secara langsung oleh pengguna akhir (*end user*) dalam kondisi yang mendekati penggunaan sebenarnya. Tujuan dari pengujian UAT yaitu untuk memastikan bahwa sistem yang dikembangkan telah berjalan sesuai dengan kebutuhan yang telah ditentukan serta layak untuk diterapkan (Durga Praveen Deevi, Naga Sushma Allur, Koteswararao Dondapati, Himabindu Chetlapalli, Thinagaran, 2024).

2.17.3 Skala Likert

Skala likert adalah metode pengukuran yang digunakan untuk memperoleh data mengenai sikap, pandangan, maupun tingkat kesetujuan responden terhadap suatu objek atau pernyataan tertentu. Metode pengukuran ini diperkenalkan oleh Rensis Likert pada tahun 1931 dan masih banyak diterapkan dalam berbagai bidang penelitian hingga saat ini. Penggunaan skala likert dilakukan dengan memberikan beberapa pilihan tingkat jawaban kepada responden terhadap pernyataan yang diberikan, sehingga peneliti dapat melakukan pengukuran dan analisis terhadap responden secara terstruktur (Simamora, 2022).

2.18 Gambaran Umum Objek Penelitian

Objek penelitian dalam penelitian ini adalah Ruang Belajar Sa'idah yang berlokasi di Kabupaten Bandung. Ruang Belajar Sa'idah merupakan lembaga pendidikan nonformal yang menyelenggarakan kegiatan pembelajaran agama Islam bagi anak-anak. Penelitian ini dilakukan karena proses pengelolaan data akademik di Ruang Belajar Sa'idah masih dilakukan secara manual sehingga diperlukan sistem informasi akademik berbasis *website* untuk membantu proses administrasi dan penyampaian informasi kepada pengajar, orang tua, maupun penanggung jawab.

2.18.1 Profil Ruang Belajar Sa'idah

Ruang Belajar Sa'idah merupakan lembaga pendidikan non formal yang bergerak di bidang pembelajaran agama Islam bagi anak-anak. Kegiatan pembelajaran meliputi berbagai materi keislaman, seperti Aqidah, Adab, Al Qur'an, Hadits, Sirah Nabawiyah, Fiqih dan Bahasa Arab. Selain pembelajaran agaman, Ruang Belajar Sa'idah saat ini juga mulai mengembangkan konsep sebagai bimbingan belajar yang mendukung proses belajar peserta didik melalui kegiatan pembelajaran yang terarah dan berkesinambungan.

Ruang Belajar Sa'idah mulai menyelenggarakan kegiatan belajar mengajar pada tahun 2017 dengan nama Madrasah Miftahussalam. Seiring perkembangan kegiatan pembelajaran serta kebutuhan masyarakat, nama tersebut kemudian berubah menjadi Ruang Belajar Sa'idah sebagai identitas lembaga yang digunakan hingga saat ini.

Dalam proses pembelajarannya, Ruang Belajar Sa'idah berpedoman pada Al Qur'an dan As-Sunnah serta berupaya menciptakan lingkungan belajar yang nyaman, kondusif, dan menyenangkan agar peserta didik dapat memahami serta mengamalkan ajaran Islam dalam kehidupan sehari-hari.

2.18.2 Visi dan Misi

Visi

Mewujudkan generasi Islam yang berakhlakul karimah, mencintai Al Qur'an dan As-Sunnah, serta mampu mengamalkan nilai-nilai Islam dalam kehidupan sehari-hari sebagai bekal meraih kebahagiaan dunia dan akhirat.

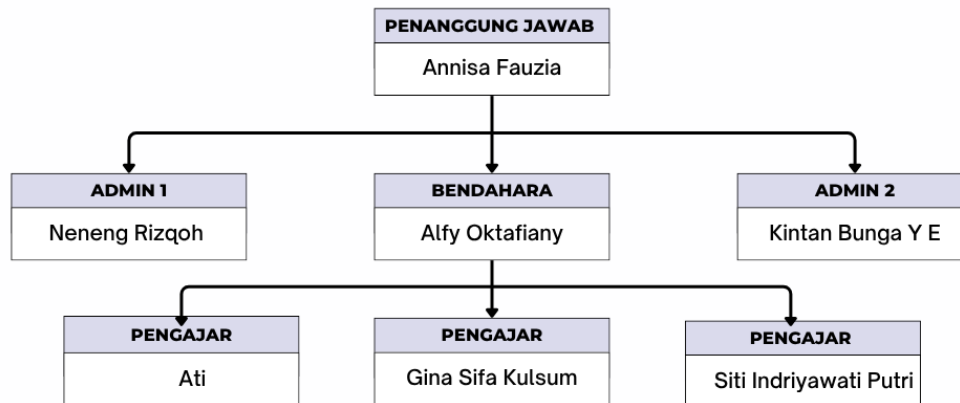
Misi

1. Menanamkan kecintaan kepada Al Qur'an dan As-Sunnah sejak usia dini.
2. Membimbing peserta didik agar mampu membaca, memahami, menghafal, serta mengamalkan Al Qur'an sesuai dengan tuntunan Islam.
3. Menanamkan akidah yang lurus serta membiasakan pelaksanaan ibadah berdasarkan Al Qur'an dan As-Sunnah.
4. Membentuk karakter peserta didik yang berakhlakul karimah, jujur, disiplin, santun, dan bertanggung jawab.
5. Menciptakan suasana belajar yang nyaman, aman, menyenangkan, dan penuh kasih sayang sehingga peserta didik memiliki semangat dalam menuntut ilmu.
6. Menjalin kerja sama yang baik dengan orang tua dalam mendukung pembinaan akhlak dan pendidikan keislaman peserta didik.

2.18.3 Struktur Organisasi



STRUKTURAL RUANG BELAJAR SA'IDAH



Gambar 2. 2 Struktural Ruang Belajar Sa'idah

2.18.4 Informasi Lembaga

Berdasarkan hasil observasi dan wawancara yang dilakukan di Ruang Belajar Sa'idah, diperoleh informasi umum terkait kondisi lembaga sebagai berikut.

No	Informasi	Keterangan
1	Jumlah Kelas	4 Kelas (A, B1, B2, dan C)
2	Jumlah Mata Pelajaran	7 Mata Pelajaran (Aqidah, Adab, Fiqih, Al Qur'an, Hadits, Bahasa Arab, Sirah Nabawiyah)
3	Jumlah Pengajar	7 Orang Pengajar Aktif
4	Jumlah Siswa	Kurang Lebih 50
5	Jenjang Studi	Anak Usia TK hingga SD kelas 6 (4 -12 Tahun)
6	Masa Studi	Tidak Terbatas, menyesuaikan kemampuan anak, namun minimal mengikuti pembelajaran selama 6 bulan (1 semester)
7	Hari Belajar	Senin – Jum'at
8	Jam Belajar	13.30 – 17.30
9	Tahun Berdiri	2017

2.18.5 Lokasi dan Waktu Penelitian

Pelaksanaan penelitian ini bertempat di Ruang Belajar Sa'idah yang berlokasi di Gang Siliwangi IV, Kampung Cimuncang, RT 001/RW 006, Desa Ciwidey, Kecamatan Ciwidey, Kabupaten Bandung, Provinsi Jawa Barat.

Penelitian ini berlangsung selama enam bulan, terhitung sejak Februari 2026 sampai dengan Juli 2026. Dalam rentang waktu tersebut, kegiatan penelitian dilakukan melalui beberapa tahapan secara berurutan, mulai dari tahap penyusunan proposal, kemudian data dikumpulkan melalui observasi dan wawancara, menganalisis kebutuhan sistem, merancang sistem, proses implementasi, pengujian sistem, hingga tahap akhir berupa penyusunan laporan tugas akhir.