

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem Informasi

2.1.1. Pengertian Sistem

Sistem adalah sekumpulan elemen atau komponen yang saling berhubungan dan berinteraksi satu sama lain untuk mencapai tujuan tertentu. Setiap komponen dalam sistem memiliki fungsi yang saling mendukung sehingga membentuk suatu kesatuan yang terorganisasi. Sistem tidak dapat berdiri sendiri, melainkan dipengaruhi oleh lingkungan sekitarnya dan bekerja berdasarkan aturan atau mekanisme tertentu yang telah ditetapkan (Sains et al., 2024).

sistem informasi merupakan sekumpulan elemen-elemen yang memiliki tujuan yang sama untuk mempertemukan kebutuhan pengolahan transaksi harian, mendukung fungsi operasi organisasi yang bersifat manajerial, serta menyediakan laporan-laporan yang diperlukan oleh pihak luar. Implementasi sistem informasi yang terintegrasi sangat penting dalam sebuah organisasi karena mampu mengolah data menjadi informasi yang lebih bermanfaat, akurat, dan tepat waktu bagi penggunanya (Fahrissal, 2018)

Secara umum, sistem terdiri atas komponen input, proses, dan output, serta dilengkapi dengan mekanisme pengendalian dan umpan balik (feedback). Input merupakan data atau sumber daya yang dimasukkan ke dalam sistem, kemudian diolah melalui suatu proses untuk menghasilkan output berupa informasi atau

hasil yang bernilai guna. Umpan balik digunakan untuk mengevaluasi kinerja sistem dan memastikan bahwa tujuan sistem dapat tercapai secara optimal (Fahrissal, 2018)

Dalam konteks sistem informasi, sistem dirancang secara terstruktur dan sistematis agar mampu mengolah data secara efektif dan efisien. Sistem yang baik harus mampu menghasilkan informasi yang akurat, relevan, dan tepat waktu untuk mendukung kegiatan operasional serta pengambilan keputusan. Oleh karena itu, pemahaman terhadap konsep sistem menjadi dasar penting dalam perancangan dan pengembangan sistem informasi berbasis teknologi.

2.1.2. Pengertian Informasi

Informasi merupakan data yang telah diolah, diklasifikasikan, atau diinterpretasikan sehingga memiliki makna dan nilai guna bagi penerimanya. Informasi berfungsi untuk mengurangi ketidakpastian dan membantu individu atau organisasi dalam memahami suatu kondisi atau permasalahan tertentu. Data yang belum diproses tidak memiliki nilai yang signifikan sebelum diubah menjadi informasi yang relevan dan dapat dipahami (Bimantoro et al., 2024).

Informasi yang berkualitas harus memenuhi beberapa kriteria utama, antara lain akurat, relevan, tepat waktu, dan lengkap. Akurasi menunjukkan tingkat kebenaran informasi, relevansi menunjukkan kesesuaian informasi dengan kebutuhan pengguna, sedangkan ketepatan waktu memastikan informasi tersedia saat dibutuhkan. Informasi yang memenuhi kriteria tersebut sangat penting dalam mendukung kegiatan operasional, pengendalian, serta pengambilan keputusan dalam suatu organisasi. Dalam konteks sistem informasi, informasi menjadi

output utama dari proses pengolahan data yang dilakukan oleh sistem. Informasi digunakan sebagai dasar dalam perencanaan, evaluasi, dan pengambilan keputusan strategis.

2.1.3. Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem terintegrasi yang terdiri dari unsur manusia, perangkat keras, perangkat lunak, basis data, dan prosedur yang saling berinteraksi untuk mengumpulkan, mengolah, menyimpan, dan menyajikan informasi yang mendukung tujuan organisasi. Konsep ini menekankan bahwa komponen-komponen tersebut bekerja bersama secara terstruktur sehingga memungkinkan aliran data menjadi informasi yang berguna bagi pengambilan keputusan (Fachri et al., 2023).

Sistem informasi dirancang untuk memenuhi kebutuhan operasional, manajerial, dan strategis organisasi dengan menghasilkan informasi yang akurat, relevan, tepat waktu, dan dapat dipercaya. Desain yang baik melibatkan pengaturan alur input–proses–output, mekanisme kontrol, serta umpan balik untuk evaluasi dan perbaikan berkelanjutan sehingga sistem tetap responsif terhadap perubahan kebutuhan pemangku kepentingan. (Prinsip-prinsip dan standar kesehatan terkait pengelolaan data juga ditekankan pada laporan global tentang sistem data kesehatan).

Dalam pelayanan posyandu, sistem informasi membantu kader dan petugas kesehatan mencatat kunjungan, memantau status gizi dan imunisasi balita, mengelola jadwal, serta menyajikan laporan bagi puskesmas dan pemangku kebijakan. Penerapan sistem informasi posyandu dapat meningkatkan efisiensi

pencatatan, akurasi data, dan kecepatan pelaporan sehingga mendukung pengambilan keputusan berbasis bukti pada tingkat layanan primer. Beberapa studi dan pengembangan sistem posyandu berbasis web/elektronik di Indonesia menunjukkan manfaat praktisnya dalam meningkatkan kualitas layanan (Dan et al., 2025).

2.2. Perancangan

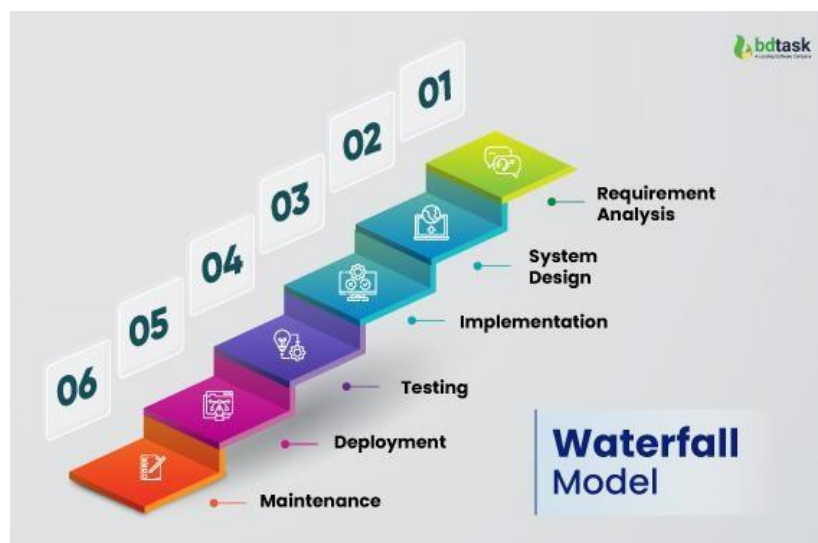
Perancangan sistem informasi pelayanan posyandu berbasis web merupakan proses penting yang digunakan untuk menciptakan suatu sistem yang dapat mengelola seluruh data kegiatan posyandu secara terstruktur, efektif, dan efisien. Dalam perancangan ini, proses perancangan bertujuan untuk mengubah kebutuhan fungsional dan non-fungsional menjadi bentuk dokumentasi desain yang siap diimplementasikan ke dalam kode program (Sugiyono et al., 2024). Tahap perancangan tidak hanya mencakup struktur teknis sistem tetapi juga alur kerja yang akan dijalankan oleh pengguna, seperti kader posyandu, petugas kesehatan, dan admin sistem. Konsep perancangan yang matang dapat membantu meminimalkan risiko kesalahan fungsi pada tahap implementasi maupun ketika sistem sudah digunakan secara langsung di lapangan.

Perancangan sistem informasi harus memperhatikan kebutuhan pengguna, alur data, dan struktur penyimpanan data sehingga seluruh proses mulai dari pencatatan data peserta, pengolahan data timbang gizi, hingga pembuatan laporan rekap dapat berjalan dengan baik. Dalam konteks pelayanan posyandu, sistem ini bertujuan untuk menggantikan pencatatan manual yang selama ini rentan terhadap kehilangan data dan ketidaktepatan input. Dengan adanya sistem berbasis web,

data yang dimasukkan dapat tersimpan langsung ke dalam database sehingga mempermudah kader dalam memantau perkembangan balita atau ibu hamil dari waktu ke waktu dan memberikan layanan kesehatan yang lebih tepat sasaran. Desain sistem ini akan menjadi dasar pengembangan aplikasi menggunakan bahasa pemrograman PHP dan basis data MySQL (Nisaa' et al., 2022).

2.2.1. Metode Waterfall

Metode Waterfall adalah salah satu model pengembangan perangkat lunak yang bersifat linier dan sekuensial, dimana setiap tahapan pengembangan mengikuti urutan tertentu dan harus selesai sebelum masuk pada tahapan berikutnya.



Gambar 2. 1 Model Waterfall

Tahapan umum dalam model *Waterfall* mencakup:

1. Tahapan Analisis Kebutuhan

Pada tahapan ini dilakukan untuk mengumpulkan dan mendokumentasikan seluruh kebutuhan sistem dari pengguna atau pemangku kepentingan.

Kebutuhan yang dikumpulkan mencakup fungsi apa saja yang diinginkan sistem, batasan sistem, serta kebutuhan fungsional dan *non-fungsional*. Tujuan utamanya adalah mendapatkan gambaran jelas mengenai apa yang harus dibangun sebelum tahap selanjutnya dimulai.

2. Tahapan Desain Sistem

Tahap desain sistem ini dilakukan untuk membuat rancangan teknis dari sistem yang akan dibangun. Ini termasuk desain arsitektur, struktur database, antarmuka pengguna, dan alur proses kerja. Output dari tahapan ini menjadi blueprint yang akan dijadikan panduan pada saat pembuatan atau pengkodean sistem.

3. Tahapan Implementasi

Tahapan implementasi, desain yang sudah dibuat diterjemahkan ke dalam kode program oleh pengembang. Untuk sistem informasi pelayanan posyandu berbasis web, implementasi ini dilakukan menggunakan bahasa pemrograman seperti PHP yang berfungsi sebagai logika aplikasi serta interaksi dengan database MySQL. Tahap ini merupakan proses “membangun sistem secara nyata”.

4. Tahapan Pengujian

Tahapan pengujian ini dilakukan untuk memastikan bahwa sistem berjalan sesuai kebutuhan yang telah ditentukan. Pengujian mencakup pemeriksaan fungsi, integrasi antar fitur, validasi data, hingga pemastian tidak adanya bug yang signifikan sebelum sistem digunakan langsung oleh pengguna.

5. Tahapan Deployment

Tahapan deployment ini merupakan fase dalam siklus pengembangan perangkat lunak di mana sistem yang telah selesai dikembangkan dan diuji kemudian diimplementasikan ke lingkungan operasional agar dapat digunakan oleh pengguna akhir. Pada tahap ini, perangkat lunak dipindahkan dari lingkungan pengembangan (development environment) ke lingkungan produksi (production environment), sehingga sistem siap digunakan secara nyata dalam mendukung aktivitas organisasi.

6. Tahapan Maintenance

Tahap pemeliharaan adalah fase setelah sistem dipakai yang berfokus pada memperbaiki error (bug), menambah atau memperbarui fitur, serta meningkatkan performa dan keamanan agar sistem tetap berjalan optimal dan relevan dengan kebutuhan pengguna. Model struktur dan alurnya yang jelas sehingga memudahkan pengembangan dalam memetakan progres pengerjaan proyek.

Dalam pengembangan sistem informasi pelayanan posyandu berbasis *web* ini, metode *Waterfall* dipilih karena kebutuhan sistem relatif sudah terdefinisi dengan jelas sejak awal. Analisis kebutuhan mencakup pengumpulan informasi tentang data apa saja yang perlu dimasukkan (misalnya data peserta, data penimbangan, imunisasi) serta hak akses pengguna. Desain sistem meliputi pembuatan diagram *UML* untuk memodelkan alur sistem sehingga pengembang dapat memahami hubungan antar komponen. Implementasi dilakukan dengan menulis kode program *PHP* dan menyiapkan database *MySQL*. Tahapan akhir

berupa pengujian untuk memastikan sistem berjalan sesuai kebutuhan dan pemeliharaan sistem untuk perbaikan di masa depan.

2.3. Pelayanan Posyandu

Pelayanan Posyandu adalah salah satu bentuk kegiatan pelayanan kesehatan masyarakat di tingkat paling dasar, yang dilaksanakan secara terpadu untuk ibu dan anak terutama balita. Posyandu memiliki tujuan utama untuk meningkatkan derajat kesehatan masyarakat melalui pemantauan pertumbuhan balita, pemberian imunisasi, penyuluhan gizi, serta pencegahan penyakit. Posyandu berperan sebagai sarana penting dalam pemantauan kesehatan ibu hamil dan balita di lingkup keluarga dan komunitas sehingga masalah kesehatan dapat terdeteksi lebih dini dan dapat ditangani segera (Kristiyanto & Pramadjaya, 2022).

Selain berfungsi sebagai tempat pelayanan kesehatan dasar, posyandu juga bertindak sebagai wadah pemberdayaan masyarakat dalam mendukung kegiatan kesehatan. Kegiatan ini biasanya melibatkan kader masyarakat yang dilatih untuk mencatat data layanan kesehatan, membantu proses timbang gizi, dan menyampaikan informasi kesehatan kepada ibu peserta. Dengan adanya sistem informasi yang terkomputerisasi, semua proses pelayanan ini dapat terekam secara tepat dan mudah dianalisis guna meningkatkan kualitas layanan posyandu secara keseluruhan (Nugraha, 2021).

2.3.1. Pemahaman Sistem informasi Pelayanan Posyandu

Sistem informasi pelayanan posyandu adalah sistem yang dirancang untuk mengelola dan menyimpan data layanan posyandu secara digital. Dalam sistem

manual, kader harus mencatat data peserta secara fisik di buku catatan yang rawan kesalahan input dan rentan hilang. Dengan menggunakan sistem informasi berbasis web, data dapat diinput langsung oleh petugas melalui aplikasi yang tersambung dengan database MySQL sehingga data tersimpan secara sistematis dan terstruktur. Hal ini memudahkan proses pencarian, pemantauan, dan pelaporan setiap waktu (Bimantoro et al., 2024).

Sistem informasi ini akan membantu dalam integrasi data dengan pihak puskesmas atau dinas kesehatan dengan cara mempermudah proses rekapitulasi laporan secara elektronik. Algoritme sistem dapat dimanfaatkan untuk membuat laporan capaian layanan seperti persentase imunisasi, status gizi balita, dan jumlah peserta kunjungan. Ini berarti petugas kesehatan dapat lebih cepat mengambil keputusan berbasis data dan merencanakan kegiatan lanjutan yang lebih efektif untuk pelayanan masyarakat.

2.3.2. Kegunaan Sistem Informasi Pelayanan Posyandu

Sistem informasi yang terintegrasi memberikan manfaat signifikan dalam pelayanan posyandu. Pertama, sistem ini memungkinkan pengelolaan data peserta secara akurat dan mudah diakses kapan saja, tanpa harus mencari data secara manual di catatan buku. Ini sangat penting dalam monitoring perkembangan balita dari waktu ke waktu karena setiap data kesehatan tersimpan dalam database dan dapat dianalisis secara cepat (Aprilya & Yulef Dian, 2025).

Sistem berbasis web dapat menghasilkan laporan otomatis yang dibutuhkan oleh puskesmas atau dinas kesehatan. Laporan ini berupa grafik pertumbuhan balita, statistik imunisasi, dan catatan pelaksanaan kegiatan

posyandu yang dapat digunakan untuk evaluasi program pelayanan Kesehatan (Aynullutfihana et al.,2025).

2.4. Rekap Data

Rekap data merupakan proses pengumpulan, pengorganisasian, dan penyajian data dalam bentuk laporan yang terstruktur dan mudah dipahami. Dalam pelayanan posyandu, proses ini mencakup pengolahan hasil timbang gizi, catatan imunisasi, hingga laporan kunjungan peserta. Dengan struktur database yang tepat, rekap data dapat disajikan secara otomatis berdasarkan periode tertentu seperti bulanan atau tahunan, sehingga petugas tidak perlu melakukan penghitungan manual yang memakan waktu dan rentan kesalahan.

Rekap data yang akurat sangat penting bagi puskesmas dan dinas kesehatan dalam menentukan kebijakan atau tindakan lanjutan, seperti fokus pada pemberian suplementasi gizi atau intensifikasi imunisasi. Penggunaan database MySQL memungkinkan data direkap dengan kemampuan query yang cepat dan efisien, serta mendukung proses ekspor laporan ke format PDF atau Excel sesuai kebutuhan instansi yang meminta.

2.5. Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk menggambarkan struktur dan perilaku sistem secara grafis. Dalam perancangan sistem informasi layanan posyandu, UML sangat penting untuk menggambarkan kebutuhan sistem dan interaksinya dengan pengguna secara jelas sebelum kode program ditulis. UML membantu semua pihak

(pengembang, pemangku kebijakan, dan pengguna akhir) memiliki pemahaman yang sama tentang sistem yang akan dibangun.

Pemodelan UML biasanya mencakup Use Case Diagram, Activity Diagram, Class Diagram, Component Diagram, Sequence Diagram, dan Deployment Diagram yang masing-masing memberikan perspektif berbeda tentang sistem. Misalnya, Use Case Diagram menunjukkan hubungan antara aktor dan fungsi sistem, sedangkan Class Diagram menunjukkan struktur data yang akan digunakan dalam database.

2.5.1. Use Case Diagram

Use Case Diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk memvisualisasikan interaksi antara aktor dengan sistem serta kebutuhan fungsional sistem secara keseluruhan. Diagram ini menggambarkan siapa saja yang berinteraksi dengan sistem dan fungsi-fungsi apa saja yang mereka lakukan untuk mencapai tujuan tertentu dalam sistem yang sedang dikembangkan. Use case diagram biasanya dibuat pada awal fase analisis kebutuhan sehingga menjadi dasar bagi perancangan sistem yang akan dibangun karena memberikan gambaran fungsionalitas dari perspektif pengguna.

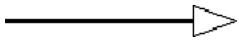
Selain menggambarkan hubungan antara aktor dan sistem, Use Case Diagram juga memudahkan tim pengembang dan pemangku kepentingan (stakeholder) dalam memahami apa yang diharapkan dari sistem tanpa harus membahas detail teknis implementasi. Dalam konteks pemodelan, diagram ini menekankan apa yang sistem lakukan (functional requirements) dari pada

bagaimana sistem melakukannya. Hal ini membuat Use Case Diagram terutama berguna pada fase awal pengembangan perangkat lunak saat kebutuhan sistem dibahas bersama dengan klien atau pengguna akhir.

Use Case Diagram juga sering dijadikan alat bantu untuk menyiapkan pengujian sistem karena setiap use case dapat berubah menjadi skenario pengujian untuk memastikan bahwa sistem berfungsi sesuai kebutuhan pengguna. Dengan memetakan use case seperti input data peserta, ubah data kader, input hasil timbang, dan generate laporan, diagram ini menyediakan gambaran struktur kebutuhan fungsional secara menyeluruh dan sistematis.

Selain itu, diagram ini dapat digunakan untuk memvalidasi kebutuhan bersama aktor yang beragam, misalnya admin, petugas posyandu, dan kader sehingga pengembang dapat memastikan bahwa setiap jenis pengguna memiliki hak akses dan fungsi yang sesuai. Misalnya, petugas posyandu mungkin hanya perlu use case input data kesehatan balita, sementara admin memiliki akses ke use case lebih luas seperti manage users dan generate laporan lengkap. Dengan demikian, Use Case Diagram tidak hanya mendeskripsikan fungsi yang ada, tetapi juga membantu dalam mendesain hak akses fungsi yang berbeda antar pengguna dalam suatu sistem.

Tabel 2. 1 Simbol Use Case

Gambar Simbol	Nama	Deskripsi
	<i>Actor</i>	Mewakili penggunaan atau entitas luar yang berinteraksi dengan sistem.
	<i>Use Case</i>	Fungsi atau layanan yang diberikan sistem.
	<i>Association</i>	Garis penghubung antara aktor dan use case.
	<i>Include</i>	Relasi yang menunjukkan bahwa satu Use Case selalu memanggil Use Case lain sebagai bagian dari prosesnya.
	<i>Extend</i>	Relasi yang menunjukkan Use Case lain yang memperluas fungsi utama secara opsional.
	<i>Generalization</i>	Relasi pewarisan antara aktor atau Use Case.

Use Case Diagram merupakan salah satu diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan interaksi antara pengguna (actor) dengan sistem yang dikembangkan. Use Case Diagram berfokus pada fungsi-fungsi yang disediakan oleh sistem serta pihak-pihak yang terlibat dalam penggunaan sistem tersebut. Simbol-simbol yang digunakan dalam Use Case Diagram memiliki makna dan fungsi masing-masing. Adapun penjelasan simbol-simbol Use Case Diagram yang digunakan pada penelitian ini dapat dijelaskan sebagai berikut:

Actor merupakan simbol yang digunakan untuk mewakili pengguna atau entitas di luar sistem yang berinteraksi secara langsung dengan sistem. Actor dapat berupa manusia, organisasi, maupun sistem lain yang memiliki peran tertentu dalam penggunaan sistem. Actor tidak termasuk bagian dari sistem, tetapi berfungsi sebagai pihak yang memicu atau menerima layanan dari sistem.

Use Case adalah simbol yang menggambarkan fungsi, layanan, atau aktivitas yang dapat dilakukan oleh sistem. Use case menunjukkan apa yang dapat dikerjakan oleh sistem sesuai dengan kebutuhan pengguna. Setiap use case merepresentasikan satu tujuan yang ingin dicapai oleh actor melalui interaksi dengan sistem.

Association merupakan garis penghubung antara actor dan use case yang menunjukkan adanya hubungan atau interaksi. Association menandakan bahwa actor memiliki hak untuk menjalankan atau menggunakan fungsi yang terdapat pada use case yang dihubungkan.

Include adalah relasi yang menunjukkan bahwa suatu use case selalu

melibatkan use case lain sebagai bagian dari prosesnya. Relasi include bersifat wajib, artinya use case yang di-include akan selalu dijalankan ketika use case utama dieksekusi. Include biasanya digunakan untuk menghindari pengulangan fungsi yang sama pada beberapa use case.

Extend merupakan relasi yang menunjukkan bahwa suatu use case dapat memperluas atau menambahkan perilaku dari use case utama. Relasi extend bersifat opsional dan hanya dijalankan pada kondisi tertentu sesuai kebutuhan sistem. Use case yang melakukan extend tidak selalu dijalankan setiap kali use case utama dijalankan.

Generalization adalah relasi pewarisan yang menunjukkan adanya hubungan umum dan khusus, baik antar actor maupun antar use case. Actor atau use case yang bersifat khusus akan mewarisi karakteristik dan perilaku dari actor atau use case yang bersifat umum. Relasi ini digunakan untuk menyederhanakan model dan menghindari duplikasi.

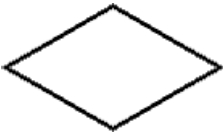
2.5.2. Activity Diagram

Activity Diagram menggambarkan alur kerja proses dalam sistem secara rinci. Diagram ini memperlihatkan tahapan proses mulai dari input data peserta, verifikasi, hingga penyimpanan ke database. Diagram alur ini memudahkan pengembang untuk memahami proses sekuensial dalam sistem serta membantu dalam pengujian kebutuhan fungsional pada setiap tahapan proses.

Dalam sistem pelayanan posyandu, Activity Diagram dapat menampilkan alur proses seperti kader melakukan pencatatan data balita, sistem melakukan validasi, kemudian memasukkan data ke tabel MySQL, dan akhirnya data siap

direkap dalam laporan.

Tabel 2. 2 Simbol Activity Diagram

Gambar Simbol	Nama Simbol	Deskripsi
	<i>Initial Node</i>	Menunjukkan titik awal dimulainya aktivitas dalam sistem.
	<i>Activity / Action</i>	Mewakili aktivitas atau proses yang dilakukan dalam alur sistem.
	<i>Decision / Marge</i>	Digunakan untuk percabangan keputusan atau penggabungan alur.
	<i>Control Flow</i>	Menunjukkan arah alur proses dari satu aktivitas ke aktivitas lain.
	<i>Final Node</i>	Menunjukkan akhir dari seluruh aktivitas dalam sistem.

Activity Diagram merupakan salah satu diagram dalam Unified Modeling Language (UML) yang digunakan untuk memodelkan alur kerja (workflow) atau

alur aktivitas suatu sistem dari awal hingga akhir. Diagram ini menggambarkan urutan aktivitas, pengambilan keputusan, serta kondisi yang terjadi dalam proses bisnis atau sistem yang dirancang. Adapun simbol-simbol yang digunakan dalam Activity Diagram beserta penjelasannya adalah sebagai berikut:

Initial Node merupakan simbol yang menunjukkan titik awal dimulainya aktivitas dalam suatu sistem. Simbol ini menandakan bahwa proses atau alur kerja mulai dijalankan dari titik tersebut sebelum berpindah ke aktivitas berikutnya.

Activity atau Action adalah simbol yang digunakan untuk mewakili aktivitas atau proses yang dilakukan dalam sistem. Setiap activity menggambarkan satu langkah kerja yang dilakukan secara berurutan sesuai dengan alur proses yang telah dirancang.

Decision/Merge merupakan simbol yang digunakan untuk percabangan keputusan atau penggabungan alur berdasarkan kondisi tertentu. Decision digunakan ketika alur proses harus memilih salah satu jalur berdasarkan suatu kondisi, sedangkan merge digunakan untuk menggabungkan kembali beberapa alur menjadi satu alur proses.

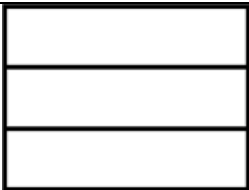
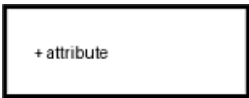
Control Flow adalah simbol berupa panah yang berfungsi untuk menunjukkan arah alur proses dari satu aktivitas ke aktivitas lainnya. Control flow menggambarkan urutan eksekusi aktivitas dalam sistem sehingga alur proses dapat dipahami secara jelas.

Final Node merupakan simbol yang menunjukkan titik akhir dari seluruh aktivitas dalam sistem. Simbol ini menandakan bahwa proses telah selesai dan tidak ada aktivitas lanjutan setelah titik tersebut.

2.5.3. Class Diagram

Class Diagram digunakan untuk memodelkan struktur kelas dalam sistem termasuk atribut dan relasinya. Class Diagram ini berhubungan langsung dengan desain basis data MySQL yang akan digunakan. Misalnya, entitas seperti Peserta, Kader, Pelayanan, dan Rekapitulasi akan diwakili sebagai tabel dalam database. Diagram ini penting dalam memastikan bahwa semua struktur data yang dibutuhkan sistem telah lengkap serta relasi antar tabel dirancang secara konsisten agar query database dapat berjalan efisien.

Tabel 2. 3 Simbol Class Diagram

Gambar Simbol	Nama Simbol	Deskripsi
	<i>Class</i>	Mewakili <i>blueprint</i> atau cetak biru dari objek dalam sistem.
	<i>Attribute</i>	Menunjukkan properti atau data yang dimiliki oleh suatu class.
	<i>Method</i>	Menunjukkan operasi atau fungsi yang dapat dilakukan oleh class.
	<i>Association</i>	Hubungan umum antara dua <i>class</i> .

	<i>Inheritance</i>	Hubungan pewarisan antara class induk dan <i>class</i> turunan.
	<i>Aggregation</i>	Hubungan keseluruhan-bagian yang bersifat lemah.

Class Diagram merupakan salah satu diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan struktur statis sistem. Diagram ini menunjukkan class, atribut, method, serta hubungan antar class yang membentuk sistem secara keseluruhan. Class Diagram berfungsi sebagai dasar dalam perancangan basis data maupun pengembangan kode program. Adapun simbol-simbol yang digunakan dalam Class Diagram pada penelitian ini dijelaskan sebagai berikut:

1. *Class*

Class merupakan simbol yang digunakan untuk merepresentasikan *blueprint* atau cetak biru dari suatu objek dalam sistem. *Class* menggambarkan kumpulan objek yang memiliki atribut dan *method* yang sama. Di dalam *class* biasanya terdapat tiga bagian utama, yaitu nama *class*, atribut, dan *method*.

2. *Attribute*

Attribute adalah simbol yang menunjukkan properti atau data yang dimiliki oleh suatu *class*. *Attribute* merepresentasikan karakteristik atau informasi

yang disimpan dalam sebuah *class* dan biasanya digunakan sebagai dasar dalam pembentukan struktur tabel pada basis data.

3. *Method*

Method merupakan simbol yang menggambarkan operasi, fungsi, atau perilaku yang dapat dilakukan oleh suatu *class*. *Method* berisi proses atau logika yang dijalankan oleh *class* untuk mengolah data yang dimilikinya.

4. *Association*

Association adalah simbol yang menunjukkan hubungan umum antara dua *class* atau lebih. *Association* menandakan bahwa *class-class* tersebut saling berhubungan dan dapat saling berinteraksi dalam sistem tanpa adanya hubungan kepemilikan secara langsung.

5. *Inheritance*

Inheritance merupakan simbol yang menggambarkan hubungan pewarisan antara *class* induk (*superclass*) dan *class* turunan (*subclass*). *Class* turunan akan mewarisi atribut dan *method* dari *class* induk sehingga dapat mengurangi duplikasi kode dan meningkatkan efisiensi perancangan sistem.

6. *Aggregation*

Aggregation adalah simbol yang menunjukkan hubungan keseluruhan dan bagian (*whole-part*) yang bersifat lemah. Pada hubungan ini, *class* bagian masih dapat berdiri sendiri meskipun *class* keseluruhan tidak ada. *Aggregation* biasanya digunakan untuk menunjukkan hubungan kepemilikan yang tidak mutlak.

2.5.4. Component Diagram

Component Diagram menunjukkan pembagian modul perangkat lunak yang akan dibangun, misalnya modul autentikasi pengguna, modul input data peserta, dan modul laporan. Diagram ini memetakan struktur fisik aplikasi sehingga memudahkan dalam proses pengembangan modul serta integrasi antar komponen.

Tabel 2. 4 Simbol Component Diagram

Gambar Simbol	Nama Simbol	Deskripsi
	<i>Component</i>	Mewakili bagian modular sistem yang menyediakan fungsi tertentu.
	<i>Interface</i>	Menunjukkan layanan atau fungsi yang disediakan atau digunakan komponen.
	<i>Dependency</i>	Menunjukkan hubungan ketergantungan antar komponen.
	<i>Port</i>	Titik interaksi antara komponen dan lingkungannya.

Component Diagram merupakan salah satu diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan struktur fisik dan arsitektur perangkat lunak. Diagram ini menunjukkan bagaimana sistem dibagi ke dalam komponen-komponen modular serta bagaimana hubungan dan ketergantungan antar komponen tersebut. *Component Diagram* sangat membantu dalam memahami pembagian fungsi sistem secara keseluruhan. Adapun simbol-simbol yang digunakan dalam *Component Diagram* pada penelitian ini dijelaskan sebagai berikut:

1. *Component*

Component merupakan simbol yang digunakan untuk merepresentasikan bagian modular dari sistem yang menyediakan fungsi tertentu. Setiap *component* dapat berupa modul program, layanan, atau subsistem yang memiliki tanggung jawab spesifik dalam sistem.

2. *Interface*

Interface adalah simbol yang menunjukkan layanan atau fungsi yang disediakan atau digunakan oleh suatu komponen. *Interface* berperan sebagai penghubung antar komponen sehingga memungkinkan terjadinya komunikasi tanpa harus mengetahui detail implementasi masing-masing komponen.

3. *Dependency*

Dependency merupakan simbol yang menunjukkan hubungan ketergantungan antar komponen. *Dependency* menandakan bahwa suatu komponen membutuhkan komponen lain untuk dapat berfungsi dengan

baik. Perubahan pada komponen yang menjadi ketergantungan dapat memengaruhi komponen yang bergantung padanya.

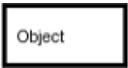
4. *Port*

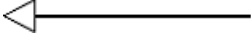
Port adalah simbol yang menggambarkan titik interaksi antara komponen dengan lingkungan luarnya. *Port* digunakan untuk menunjukkan tempat masuk dan keluarnya komunikasi atau layanan yang dimiliki oleh suatu komponen, sehingga memperjelas mekanisme interaksi antar komponen.

2.5.5. **Sequence Diagram**

Sequence Diagram memperlihatkan urutan pesan yang dikirimkan antar objek dalam sistem ketika suatu fungsi dijalankan. Contohnya, ketika petugas posyandu menekan tombol simpan data peserta, *Sequence Diagram* menunjukkan alur permintaan dari *UI*, proses validasi data, hingga penyimpanan ke *database*.

Tabel 2. 5 Sequence Diagram

Gambar Simbol	Nama Simbol	Deskripsi
	<i>Actor / Object</i>	Mewakili aktor atau objek yang terlibat dalam interaksi.
	<i>Lifeline</i>	Garis vertikal yang menunjukkan keberadaan objek selama proses berlangsung.

	<i>Activation</i>	Menunjukkan periode waktu <i>objek</i> sedang aktif memproses pesan.
	<i>Message</i>	Panah yang menunjukkan pesan atau permintaan antar <i>objek</i> .
	<i>Return Message</i>	Panah balikan yang menunjukkan hasil atau respon dari pesan.

Sequence Diagram merupakan salah satu diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan urutan interaksi antar aktor dan objek dalam sistem berdasarkan waktu. Diagram ini menekankan pada bagaimana pesan dikirim dan diterima antar objek secara berurutan untuk menjalankan suatu proses. Adapun simbol-simbol yang digunakan dalam *Sequence Diagram* pada penelitian ini dijelaskan sebagai berikut:

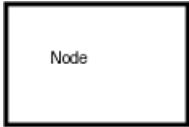
1. Actor/Object merupakan simbol yang digunakan untuk merepresentasikan aktor atau objek yang terlibat dalam suatu interaksi. Actor atau object berperan sebagai pihak yang mengirim maupun menerima pesan dalam proses yang sedang berlangsung.
2. Lifeline adalah simbol berupa garis vertikal yang menunjukkan keberadaan suatu objek selama proses interaksi berlangsung. Lifeline menggambarkan waktu hidup objek sejak awal hingga akhir interaksi pada sequence diagram.

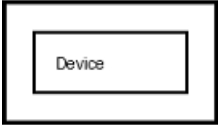
3. Activation merupakan simbol yang menunjukkan periode waktu di mana suatu objek sedang aktif dalam memproses pesan atau menjalankan operasi tertentu. Activation biasanya ditampilkan dalam bentuk persegi panjang tipis pada lifeline.
4. Message adalah simbol berupa panah yang menunjukkan pesan atau permintaan yang dikirim dari satu objek ke objek lain. Message menggambarkan komunikasi antar objek yang menyebabkan terjadinya suatu aktivitas atau proses dalam sistem.
5. Return Message merupakan simbol berupa panah balikan yang menunjukkan hasil, respon, atau nilai yang dikembalikan setelah suatu pesan diproses. Return message menandakan bahwa proses yang diminta telah selesai dijalankan.

2.5.6. Deployment Diagram

Deployment Diagram menggambarkan konfigurasi fisik perangkat keras dan lunak yang dipakai. Misalnya aplikasi akan dijalankan pada server lokal menggunakan *XAMPP (Apache + MySQL + PHP)* dan diakses oleh klien melalui browser internet.

Tabel 2. 6 Deployment Diagram

Gambar Simbol	Nama Simbol	Deskripsi
	<i>Node</i>	Mewakili <i>node</i> fisik atau logis tempat komponen dijalankan.

	<i>Device</i>	<i>Node</i> fisik berupa perangkat keras seperti server atau komputer.
	<i>Execution Environment</i>	Lingkungan tempat <i>artifact</i> dijalankan, seperti OS atau web server.
	<i>Artifact</i>	File atau aplikasi yang <i>dideploy</i> pada <i>node</i> .

Deployment Diagram merupakan salah satu diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan arsitektur fisik sistem serta bagaimana perangkat lunak diimplementasikan pada perangkat keras. Diagram ini menunjukkan hubungan antara node, device, dan artifact yang digunakan dalam proses implementasi sistem. Adapun simbol-simbol yang digunakan dalam Deployment Diagram pada penelitian ini dijelaskan sebagai berikut:

1. *Node* merupakan simbol yang digunakan untuk merepresentasikan *node* fisik atau logis sebagai tempat dijalkannya komponen sistem. *Node* dapat berupa server, komputer, maupun lingkungan logis tempat aplikasi berjalan.

2. *Device* adalah simbol yang menggambarkan *node* fisik berupa perangkat keras, seperti server, komputer, atau perangkat jaringan lainnya. *Device* berfungsi sebagai media utama untuk menjalankan sistem dan menampung komponen perangkat lunak.
3. *Execution Environment* merupakan simbol yang menunjukkan lingkungan eksekusi tempat *artifact* dijalankan. Lingkungan ini dapat berupa sistem operasi, web server, application server, atau *runtime environment* lainnya yang mendukung jalannya aplikasi.
4. *Artifact* adalah simbol yang merepresentasikan file, modul, atau aplikasi yang di *deploy* pada suatu *node*. *Artifact* dapat berupa file program, library, atau basis data yang digunakan oleh sistem dalam proses operasional.

2.6. Hypertext Preprocessor (PHP)

Hypertext Preprocessor (PHP) adalah bahasa *scripting server-side* yang dirancang untuk pengembangan aplikasi *web*. *PHP* mampu menerima *input* dari formulir *web*, berkomunikasi dengan *database MySQL*, serta menghasilkan konten *web* dinamis yang interaktif. Sebagai bahasa pemrograman yang banyak digunakan pada aplikasi *web* berbasis *open source*, *PHP* menyediakan pustaka fungsi yang lengkap dan dukungan komunitas yang luas sehingga memudahkan pengembang dalam menyelesaikan proyek kompleks seperti sistem pelayanan posyandu berbasis *web*.

Dalam sistem ini, *PHP* bertanggung jawab atas logika pengolahan data, termasuk validasi *input*, eksekusi *query* ke *database MySQL*, pengaturan hak akses pengguna, serta menghasilkan tampilan halaman *web* sesuai dengan kebutuhan

pengguna. Keunggulan *PHP* meliputi portabilitas di berbagai platform serta integrasi yang baik dengan server *Apache*, sehingga menjadikannya pilihan populer untuk pengembangan aplikasi berbasis *web*.

2.7. Pengertian Database

Database merupakan kumpulan data yang disimpan secara terstruktur dalam format digital agar dapat diakses, dikelola, dan diperbarui dengan mudah. Dalam sistem informasi pelayanan posyandu, *database* berfungsi sebagai tempat penyimpanan semua data peserta, hasil pemeriksaan gizi, catatan layanan, dan laporan rekapitulasi.

Database dirancang sedemikian rupa agar mampu menangani seluruh kebutuhan penyimpanan data secara efisien dengan mempertimbangkan normalisasi, relasi antar tabel, serta integritas data. Basis data yang dirancang dengan baik dapat meningkatkan performa *query* dan mendukung fungsi sistem seperti pencarian data, *update*, *delete*, serta pengelompokan data secara cepat.

2.7.1. XAMPP

XAMPP adalah paket perangkat lunak yang digunakan untuk membuat lingkungan *server* lokal yang mencakup *Apache HTTP Server*, *MariaDB/MySQL*, *PHP*, dan *Perl*. *XAMPP* memudahkan pengembang dalam membangun aplikasi *web* secara lokal sebelum di-*deploy* ke *server* produksi karena tidak memerlukan konfigurasi yang kompleks dan dapat dijalankan pada berbagai sistem operasi.

Penggunaan *XAMPP* dalam perancangan sistem informasi pelayanan posyandu memungkinkan pengujian aplikasi secara menyeluruh di lingkungan lokal yang mirip dengan *server* sebenarnya. *XAMPP* juga menyediakan

phpMyAdmin yang mempermudah pengelolaan *database MySQL* melalui antarmuka grafis.



Gambar 2. 2 XAMPP

2.7.2. MySQL

MySQL adalah sistem manajemen basis data relasional (*Relational Database Management System – RDBMS*) yang paling populer dan banyak digunakan di dunia, terutama untuk aplikasi berbasis *web*. *MySQL* menggunakan *Structured Query Language (SQL)* sebagai bahasa untuk menyimpan, mengelola, dan mengambil data secara terstruktur dalam bentuk tabel yang saling berelasi. Sistem ini bersifat *open-source*, sehingga dapat dimanfaatkan secara gratis dan dimodifikasi sesuai kebutuhan pengembangan aplikasi. *MySQL* bekerja dengan model *client-server*, yang memungkinkan aplikasi klien seperti *PHP* untuk mengirimkan perintah *SQL* yang kemudian diproses oleh *server database* untuk operasi penyimpanan dan pengambilan data.

Dalam konteks pengembangan sistem informasi seperti sistem pelayanan posyandu, *MySQL* menjadi pilihan yang tepat karena kemampuannya dalam menangani data terstruktur secara efisien. *MySQL* memiliki performa yang cepat,

reliabel, dan mudah digunakan, bahkan untuk aplikasi yang dihosting pada *web server* dengan beban trafik rendah hingga menengah. Selain itu, *MySQL* kompatibel dengan berbagai bahasa pemrograman populer, termasuk *PHP*, yang merupakan bahasa *server-side* utama dalam sistem yang dikembangkan. Kombinasi *PHP* dan *MySQL* menjadi solusi efektif untuk membangun aplikasi dinamis yang memerlukan pengelolaan data seperti data peserta, data kader, pelayanan, dan laporan rekapitulasi secara *real-time*.

MySQL juga menawarkan keunggulan berupa dukungan *multi-platform*, kemampuan bekerja pada berbagai sistem operasi, serta dukungan komunitas yang luas yang membantu *developer* dalam menyelesaikan permasalahan pengembangan. Fitur-fitur seperti *query* yang teroptimasi, *indexing*, dan *caching* menjadikan *MySQL* efisien baik untuk tahap prototipe maupun produksi sistem informasi yang lebih kompleks. Selain itu, aspek keamanan data melalui kontrol akses dan kemampuan *backup* menjadikan *MySQL* layak digunakan dalam sistem yang menuntut integritas serta ketersediaan data jangka panjang.

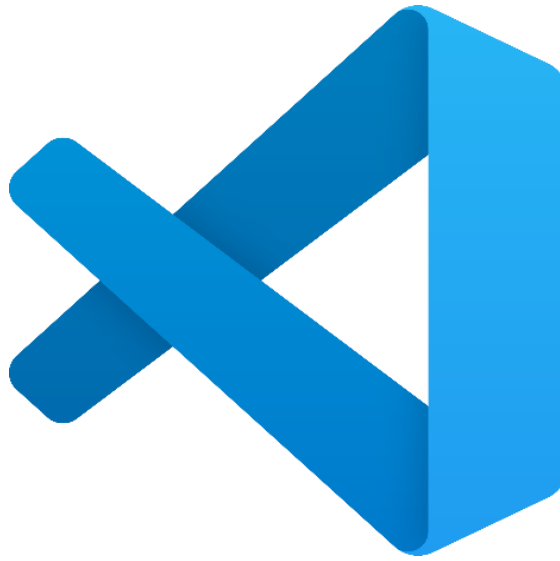
Berdasarkan alasan tersebut, seperti kompatibilitas tinggi dengan *PHP*, kemudahan penggunaan, dukungan komunitas yang luas, serta performa yang baik, *MySQL* dipilih sebagai basis data dalam pengembangan sistem informasi pelayanan posyandu berbasis *web*. Keunggulan ini mendukung operasi *CRUD* (*Create, Read, Update, Delete*) pada berbagai entitas data dalam sistem serta memberikan fleksibilitas bagi pengembang dalam mengatur struktur *database* dan mengoptimalkan performa aplikasi sesuai kebutuhan pengguna.

2.8. Tools yang Digunakan

2.8.1. Visual Studio Code

Visual Studio Code (VS Code) adalah editor kode sumber yang banyak digunakan oleh pengembang *web* karena ringan, mudah digunakan, dan kaya fitur. Tools ini dikembangkan oleh Microsoft dan mendukung berbagai bahasa pemrograman, termasuk *PHP*, *HTML*, *CSS*, dan *JavaScript*, sehingga sangat cocok untuk mengembangkan aplikasi *web* dinamis seperti sistem pelayanan posyandu. *VS Code* menyediakan berbagai fitur seperti *syntax highlighting*, *auto-completion*, *debugger*, serta dukungan *extension (plugin)* yang dapat meningkatkan produktivitas pengembang, misalnya ekstensi *PHP IntelliSense* untuk membantu melengkapi kode secara otomatis.

Dalam konteks pengembangan aplikasi *PHP* dan *MySQL*, *Visual Studio Code* menyediakan lingkungan pengembangan yang fleksibel dan dapat dikustomisasi. Pengembang dapat mengintegrasikan *terminal*, *version control* seperti *Git*, serta melakukan *debugging* langsung melalui editor. Fitur-fitur tersebut membantu pengembang dalam menulis, menguji, dan memperbaiki kode dengan lebih cepat tanpa perlu berpindah ke aplikasi lain.



Gambar 2. 3 Visual Studio Code

2.9. Penelitian Terdahulu

Penelitian yang dilakukan oleh Pamungkas (2025) mengkaji perancangan sistem informasi posyandu berbasis *web* yang bertujuan membantu kader posyandu dalam melakukan pencatatan kegiatan, *monitoring* balita, serta penyusunan laporan secara digital. Penelitian ini menggunakan metode *Waterfall* dalam pengembangan sistem. Hasil penelitian menunjukkan bahwa sistem yang dirancang mampu menggantikan proses pencatatan manual yang sebelumnya memerlukan waktu lama dan rentan terhadap kesalahan, sehingga dapat meningkatkan efisiensi kerja kader posyandu.

Selanjutnya, Atmaja (2025) berfokus pada pengembangan *Information System for Monitoring Posyandu Activities* berbasis *web* yang dirancang untuk mempermudah proses perekaman kunjungan, aktivitas, serta pengolahan data posyandu secara digital. Hasil penelitian menunjukkan bahwa penerapan sistem

berbasis *web* mampu membuat proses administrasi dan *monitoring* kegiatan posyandu menjadi lebih terstruktur, efektif, dan cepat dibandingkan dengan metode pencatatan manual.

Penelitian Aprilya dan Dian (2025) mengkaji implementasi sistem informasi posyandu digital berbasis web menggunakan PHP dan MySQL di Posyandu Nusaindah II. Hasil penelitian menunjukkan bahwa sistem mampu mempercepat pencatatan, mempermudah pengelolaan data ibu hamil dan balita, serta meningkatkan akses informasi bagi kader dan petugas posyandu.

Penelitian oleh Yanti dan Ramayanti (2025) mengkaji penerapan aplikasi posyandu berbasis *web* yang difokuskan pada *monitoring* gizi balita sebagai upaya pencegahan stunting. Hasil penelitian menunjukkan bahwa sistem berbasis *web* mampu meningkatkan kestabilan pencatatan data gizi balita dan menyediakan antarmuka yang lebih efisien untuk mendukung pelayanan kesehatan masyarakat.

Selanjutnya, Probonegoro (2025) dalam penelitiannya terhadap sistem informasi posyandu di Desa Cit menyimpulkan bahwa penggunaan sistem informasi berbasis *web* secara signifikan mempermudah pencatatan layanan kesehatan dasar serta administrasi data posyandu. Sistem ini mampu menggantikan proses manual yang sebelumnya kurang efisien dan berpotensi menimbulkan kesalahan pencatatan.

Penelitian yang dilakukan oleh Sintiya dkk. (2025) mengembangkan sistem informasi manajemen posyandu untuk mengintegrasikan data layanan kesehatan ibu dan balita. Pengembangan sistem dilakukan menggunakan metode *Extreme Programming* dengan evaluasi langsung oleh pengguna. Hasil penelitian

menunjukkan bahwa sistem yang dikembangkan telah memenuhi kebutuhan pengguna dan direkomendasikan untuk diimplementasikan secara lebih luas karena kemampuannya dalam mempermudah pengolahan dan pengelolaan data layanan kesehatan.

Penelitian oleh Fachri, Hendry, dan Zen (2023) merancang sistem informasi posyandu ibu dan anak berbasis *web* dengan memanfaatkan diagram *UML* dalam tahap perancangan sistem. Hasil penelitian menunjukkan bahwa sistem yang dibangun mampu mempermudah pengelolaan data layanan kesehatan ibu dan anak secara terpusat serta membantu kader posyandu dalam menjalankan tugas operasional secara lebih efektif.

Selanjutnya, Sukoco (2022) melakukan penelitian terkait perancangan sistem informasi posyandu berbasis *web* di Desa Tulus Rejo dengan menggunakan metode *Waterfall* serta pengujian sistem menggunakan teknik *Black Box Testing*. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu membantu kader posyandu dalam pencatatan data, pembuatan laporan, serta penyajian informasi secara digital.

Terakhir, penelitian oleh Pratama (2024) mengenai sistem informasi posyandu berbasis *web* untuk layanan kesehatan ibu dan anak menunjukkan bahwa penerapan sistem digital mampu meningkatkan akurasi data, mempercepat proses pembuatan laporan, serta mendukung pengambilan keputusan dibandingkan dengan metode pencatatan manual.

Tabel 2. 7 Penelitian Terdahulu

Peneliti	Judul Penelitian	Tahun	Data & Metode yang Digunakan	Hasil Penelitian
Pamungkas	Perancangan Sistem Informasi Posyandu Berbasis Web	2025	Data posyandu; <i>Metode Waterfall</i>	Menggantikan pencatatan manual dan meningkatkan efisiensi kerja kader
Atmaja	<i>Information System for Monitoring Posyandu Activities</i>	2025	Data kegiatan posyandu; Sistem berbasis web	Administrasi dan <i>monitoring</i> kegiatan menjadi lebih cepat dan terstruktur
Aprilya & Dian	Sistem Informasi Posyandu Digital Berbasis Web	2025	<i>PHP & MySQL</i> ; Wawancara dan Observasi	Mempercepat pencatatan dan pengelolaan data ibu dan balita
Yanti & Ramayanti	Aplikasi Posyandu Berbasis Web Untuk <i>Monitoring</i> Gizi Balita	2025	Data gizi balita; Aplikasi web	Mendukung <i>monitoring</i> gizi dan pencegahan Stunting
Probonegoro	Sistem Informasi Posyandu Berbasis Web Desa Cit	2025	Data layanan posyandu; Sistem informasi web	Mempermudah pencatatan dan administrasi layanan kesehatan
Sintiya dkk.	Sistem Informasi Manajemen Posyandu Harmoni	2025	<i>Extreme Programming</i> ; Evaluasi pengguna	Sistem memenuhi kebutuhan pengguna dan direkomendasikan implementasi luas
Fachri, Hendry & Zen	Sistem Informasi Posyandu Ibu dan Anak	2023	<i>UML</i> ; Sistem berbasis web	Pengelolaan data terpusat dan mendukung operasional kader

	Berbasis Web			
Sukoco	Perancangan Sistem Informasi Posyandu Desa Tulus Rejo	2022	Waterfall; Black Box Testing	Mempermudah pencatatan data dan pembuatan laporan
Pratama	Sistem Informasi Posyandu Berbasis Web	2024	<i>PHP</i> & <i>MySQL</i> ; Sistem informasi	Meningkatkan akurasi data dan mempercepat penyusunan laporan