

BAB II

LANDASAN TEORI

2.1. Pengertian Sistem Informasi

Sistem informasi adalah serangkaian mekanisme yang saling terhubung untuk mengelola data mulai dari pengumpulan hingga penyajian kepada pengguna akhir. Dalam konteks operasional pemerintahan desa, pengertian ini mencakup pengembangan profil desa berbasis web, digitalisasi pembukuan *BUMDes*, layanan surat elektronik, serta portal layanan publik yang memungkinkan adanya komunikasi dua arah antara pemerintah desa dan masyarakat (Sucipto dkk., 2022).

Kajian ini menyajikan ringkasan ilmiah mengenai konsep dasar sistem informasi, definisi informasi dan sistem informasi, serta fungsi sistem informasi dalam konteks pemerintahan desa. Seluruh pernyataan didukung oleh temuan penelitian dan laporan implementasi yang relevan pada konteks digitalisasi dan layanan publik di berbagai desa yang ada di Indonesia (Karimullah dkk., 2022).

2.1.1. Konsep Dasar Sistem Informasi

Sistem informasi (SI) dapat diartikan sebagai suatu rangkaian komponen teknologi, sumber daya manusia, prosedur, dan data yang berfungsi secara terintegrasi untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi guna mendukung pengambilan keputusan serta operasional organisasi. Dalam konteks pemerintahan desa, komponen-komponen tersebut meliputi:

1. Infrastruktur teknologi, seperti situs web, aplikasi mobile, dan platform OpenSID.

2. Sumber daya manusia, yang mencakup perangkat desa dan tenaga teknis.
3. Data administratif dan publik, termasuk profil desa, data penduduk, serta laporan keuangan.
4. Prosedur tata kelola dan layanan publik yang diatur melalui modul sistem dan desain arsitektur TI.

Metodologi rekayasa sistem informasi, seperti model *waterfall* atau pendekatan arsitektur *enterprise*, sering digunakan untuk merancang, mengembangkan, dan mengintegrasikan komponen-komponen tersebut agar sistem yang dibangun dapat beroperasi dengan stabil dan sesuai dengan kebutuhan pemerintahan desa (Wicaksono dkk., 2022).

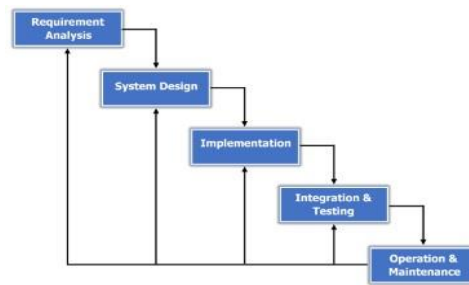
2.1.2. Peran Sistem Informasi Dalam Pemerintahan Desa

Berbagai penelitian empiris dan laporan pendampingan masyarakat menekankan peran penting sistem informasi dalam meningkatkan tata kelola pemerintahan desa. Sistem informasi terbukti meningkatkan efisiensi administrasi dengan mempercepat layanan berbasis web yang mengurangi interaksi langsung, mempercepat pengelolaan data penduduk, serta memperbaiki proses pengarsipan dokumen (Kusuma dkk., 2023). Digitalisasi pencatatan keuangan desa dan *BUMDes* memberikan dampak signifikan terhadap peningkatan efisiensi serta akurasi data. Selain itu, transparansi informasi publik juga meningkat melalui penyediaan situs web desa dan aplikasi informasi publik yang memudahkan warga mengakses profil desa, peraturan, serta layanan administrasi (Sucipto dkk., 2022). Implementasi *OpenSID* bahkan berhasil memperpendek jarak dan waktu dalam penyampaian informasi antara pemerintah desa dan masyarakat.

Sistem informasi juga berperan penting dalam mendukung penyusunan kebijakan dan perencanaan berbasis data. Penerapan arsitektur *enterprise* dan perancangan sistem yang baik memungkinkan data desa digunakan untuk merancang program, mengevaluasi kegiatan, serta memantau proyek pembangunan dengan cara yang lebih sistematis dan terukur (Wicaksono dkk., 2022). Selain itu, peningkatan kualitas layanan publik tidak hanya bergantung pada teknologi, tetapi juga pada penguatan kapasitas sumber daya manusia (SDM) desa melalui pelatihan dan pendampingan yang memungkinkan perangkat desa untuk mengelola sistem informasi secara mandiri. Dokumentasi digital juga mempermudah proses audit serta memperkuat akuntabilitas pemerintah desa, meskipun efektivitasnya tetap bergantung pada kelengkapan data, kemampuan SDM, dan kecocokan teknologi dengan kebutuhan lokal.

2.2. Metode Perancangan Sistem

Model *waterfall* adalah model yang paling sering digunakan dalam tahap pengembangan. Model *waterfall* ini juga dikenal sebagai model tradisional atau model klasik. Model *waterfall* sering disebut juga sebagai model sekuensial linier (*sequential linear*) atau alur hidup klasik (*Classic cycle*). Model *waterfall* ini menawarkan pendekatan alur hidup perangkat lunak secara sekuensial terurut yang dimulai dari analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan.



Gambar 2. 1 Metode *Waterfall*

Sumber: (Harjono & Tute, 2022)

Model *Waterfall* terdiri atas beberapa tahapan utama, yaitu:

1. Requirement Analysis

Tahap analisis kebutuhan untuk mengidentifikasi kebutuhan pengguna, kebutuhan sistem, serta masalah yang akan diselesaikan.

2. System Design

Tahap perancangan sistem, meliputi desain alur kerja, struktur sistem, antarmuka, basis data, dan kebutuhan perangkat lunak.

3. Implementation

Tahap penerjemahan hasil desain ke dalam bentuk kode program atau pembangunan aplikasi.

4. Integration & Testing

Tahap penggabungan seluruh komponen sistem dan pengujian untuk memastikan sistem berjalan sesuai kebutuhan.

5. Operation & Maintenance

Tahap penggunaan sistem secara nyata serta pemeliharaan untuk memperbaiki kesalahan, meningkatkan kinerja, atau menyesuaikan sistem dengan kebutuhan baru.

2.3. Bahasa Pemograman

Penerapan bahasa pemrograman dan komponen utama dalam pengembangan aplikasi web, seperti *PHP*, *Bootstrap*, *MySQL XAMPP*, dan *Visual Studio Code*, dijelaskan berdasarkan sintesis dari berbagai studi kasus dan publikasi ilmiah, sehingga setiap paparan mendapat dukungan referensi empiris (Elizabeth & Febriansyah, 2022). Tujuan utama dari penjelasan ini adalah untuk memberikan pemahaman teknis dan praktis mengenai peran masing-masing komponen, hubungan, dan kolaborasi di antara mereka dalam keseluruhan siklus pengembangan sistem, serta contoh penggunaan yang tercatat dalam literatur.

2.3.1. PHP

Peran *PHP* dalam Aplikasi Web Sistem Informasi adalah sebagai bahasa server-side yang mengatur logika aplikasi, memproses data, dan menghasilkan keluaran *HTML* dinamis untuk ditampilkan di sisi klien. Berbagai penelitian tentang konstruksi sistem informasi berbasis web menunjukkan bahwa *PHP* secara konsisten menjadi bahasa utama di lapisan server karena kemampuannya dalam mengelola transaksi dan berinteraksi langsung dengan basis data. Sebagai contoh, penelitian mengenai sistem kepegawaian memanfaatkan *PHP* untuk mengelola proses bisnis dan menghubungkan dengan *MySQL* sebagai sistem penyimpanan data (Darmanta Sukrianto & Sinta Maria, 2022). Hal serupa juga terlihat pada

aplikasi pemesanan produk dan repository tugas akhir yang menggunakan *PHP* untuk fungsi bisnis dan integrasi basis data.



Gambar 2. 2 PHP

2.3.2. Bootstrap

Bootstrap adalah framework CSS open source yang dirancang untuk mendukung desain responsif dalam pengembangan aplikasi web. Hal ini memungkinkan pengembang untuk membuat tampilan yang dapat beradaptasi dengan berbagai ukuran layar dan perangkat, sehingga meningkatkan pengalaman pengguna di berbagai platform. Framework ini memudahkan pengembang dalam menyusun antarmuka pengguna (*UI*) yang konsisten dan profesional dengan pengaturan yang lebih sederhana.

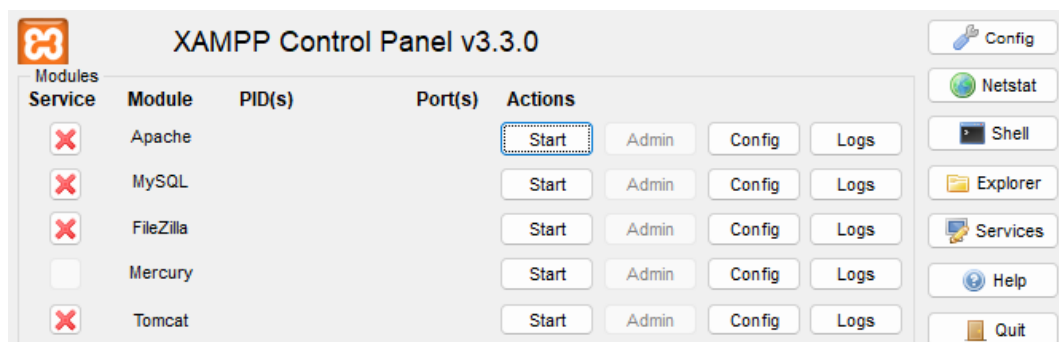
Bootstrap berperan penting dalam menyediakan komponen *UI* yang siap pakai. Fitur-fitur seperti tombol, formulir, dan navigasi (*navbar*) sudah dirancang sebelumnya dan siap digunakan, sehingga mempercepat proses pengembangan. Selain itu, pendekatan *mobile-first* dalam *Bootstrap* memastikan bahwa desain aplikasi web dapat digunakan dengan optimal pada perangkat mobile, memberikan pengalaman yang baik di berbagai ukuran layar (Putra & Maslihah, 2023).



Gambar 2. 3 Bootstrap

2.3.3. MySQL (XAMPP)

MySQL adalah sistem manajemen basis data relasional yang menyimpan data aplikasi, mulai dari data pengguna hingga riwayat transaksi. Penelitian tentang aplikasi web menunjukkan bahwa *MySQL* sering dipilih sebagai basis data utama, terutama ketika digabungkan dengan lingkungan pengembangan lokal seperti *XAMPP*. Dalam sistem repository tugas akhir, manajemen kepegawaian, hingga aplikasi *e-commerce*, *MySQL* digunakan untuk menyimpan data terstruktur yang mendukung pencarian, rekap, dan pelaporan (Darmanta Sukrianto & Sinta Maria, 2022).

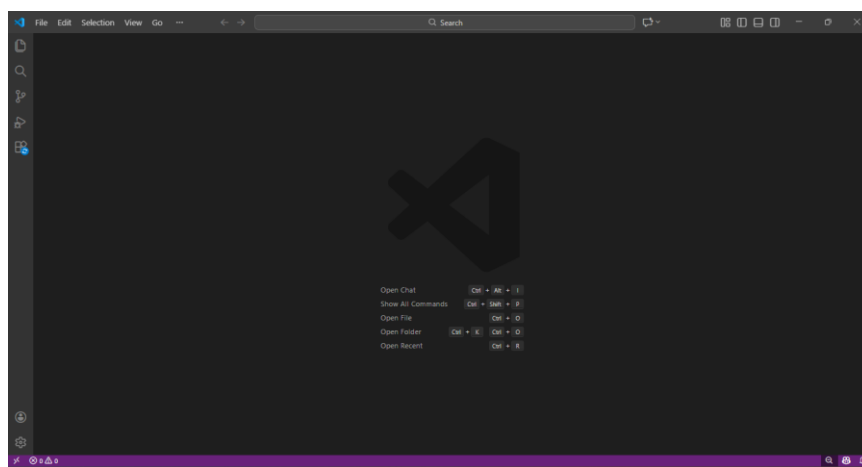


Gambar 2. 4 XAMPP

2.3.4. Visual studio Code

Dalam dunia pengembangan aplikasi web, *VS Code* sangat berguna karena mendukung berbagai bahasa pemrograman, termasuk *PHP*, *HTML*, *CSS*, dan *JavaScript*. Kemampuannya untuk berintegrasi dengan sistem kontrol versi seperti

Git sangat penting dalam kolaborasi tim, yang berkontribusi pada keberhasilan proyek perangkat lunak. Fitur debugging yang ada di dalam *VS Code* membantu pengembang dalam menemukan dan memperbaiki kesalahan dalam kode mereka. Fitur-fitur tambahan, termasuk pengelolaan versi, menjadikannya pilihan ideal untuk pengembangan aplikasi berbasis web yang kompleks, di mana keandalan dan kecepatan sangat dibutuhkan (Gayatri Makrand Ghodke & Trupti Chavan, 2024).



Gambar 2. 5 Visual Studio Code

2.4. Perancangan Sistem UML

Pendekatan perancangan sistem menggunakan *Unified Modeling Language (UML)* adalah praktik yang mengubah kebutuhan fungsional dan struktural perangkat lunak menjadi representasi visual yang terstruktur. Berbagai penelitian mengenai pengembangan aplikasi, baik berbasis web maupun mobile, menunjukkan bahwa *UML* digunakan untuk mendukung seluruh tahap pengembangan, mulai dari analisis, perancangan, implementasi, hingga pengujian. Artefak yang umumnya dihasilkan meliputi *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*, yang semuanya membantu memetakan kebutuhan sistem, struktur objek, serta interaksi dinamis dalam proses bisnis.

Penggunaan *UML* juga diintegrasikan dengan berbagai metodologi seperti *Waterfall*, sebagaimana dijelaskan dalam beberapa studi. Temuan penelitian pada sistem informasi dan aplikasi mobile semakin menguatkan posisi *UML* sebagai alat perancangan utama (Khotimah, 2024).

2.4.1. Use Case Diagram

Use Case Diagram menggambarkan bagaimana aktor, baik pengguna manusia maupun sistem eksternal, berinteraksi dengan sistem melalui serangkaian *use case* yang mewakili kebutuhan utama. Diagram ini memetakan batas sistem, peran aktor, serta layanan atau fungsi apa yang harus disediakan untuk memenuhi kebutuhan tersebut. Diagram ini digunakan untuk mengidentifikasi aktor (misalnya mahasiswa, admin, pelanggan) dan *use case* yang mencakup akses informasi, pemesanan, hingga pengelolaan data produk. Selain itu, *Use Case Diagram* berfungsi sebagai dokumentasi awal kebutuhan yang memudahkan kolaborasi antara pengembang, *product owner*, dan pemangku kepentingan dalam menyepakati ruang lingkup sistem sebelum perancangan rinci dilakukan (Afiansyah dkk., 2023).

Tabel 2. 1 Use Case Diagram

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
3		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan

			struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
5		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
6		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

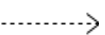
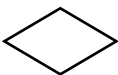
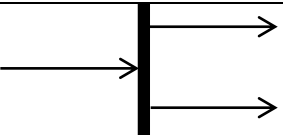
Sumber: (Harlina, 2025)

2.4.2. Activity Diagram

Activity Diagram adalah diagram perilaku yang menggambarkan aliran aktivitas atau langkah-langkah dalam suatu proses bisnis atau *use case*. Diagram ini berfokus pada alur kendali (*control flow*), percabangan, penggabungan, serta aktivitas paralel (*fork/join*), sehingga memungkinkan analisis mendalam mengenai proses yang sedang. Dalam penerapannya, diagram ini sering digunakan untuk menggambarkan alur proses seperti registrasi dan pembayaran daring, pemesanan makanan, atau pengelolaan transaksi keuangan, di mana langkah-langkah dari awal hingga akhir digambarkan secara sistematis (Mulyani & Sulastri, 2022).

Kegunaan diagram ini terutama terletak pada kemampuannya untuk memperjelas proses operasional, mengidentifikasi potensi hambatan atau kebutuhan *exception handling*, serta menjadi panduan dalam implementasi logika sistem, baik di sisi antarmuka maupun layanan belakang (*backend*), seperti proses *checkout* atau verifikasi pembayaran. Studi penerapan *UML* pada berbagai domain juga menunjukkan bahwa *Activity Diagram* sangat efektif untuk menerjemahkan proses bisnis menjadi desain perangkat lunak yang dapat diotomatisasi.

Tabel 2. 2 Activity Diagram

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Menyatakan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Control Flow</i>	Menunjukkan Urutan Eksekusi.
3		<i>Object Flow</i>	Menunjukkan aliran objek dari sebuah action atau activity ke action.
4		<i>Initial Node</i>	Menyatakan bahwa sebuah objek dibentuk atau diawali.
5		<i>Activity Final Node</i>	Menyatakan bahwa sebuah objek dibentuk atau diakhiri.
6		<i>Decision</i>	Menunjukkan penggambaran suatu keputusan/tindakan yang harus diambil pada kondisi tertentu.
7		<i>Fork</i>	Menyatakan untuk memecah behavior menjadi activity atau action yang parallel.

Sumber: (Harlina, 2025)

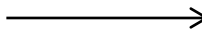
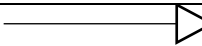
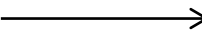
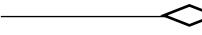
2.4.3. Class Diagram

Class Diagram adalah representasi struktur statis sistem yang menggambarkan kelas, atribut, operasi, serta hubungan antar kelas seperti asosiasi, agregasi, komposisi, dan pewarisan. Diagram ini menjadi dasar pemodelan berorientasi objek yang selanjutnya diterjemahkan menjadi desain basis data, model domain, atau struktur *API* dalam pengembangan perangkat lunak. Dalam berbagai penelitian, seperti perancangan sistem penjualan, pengelolaan kas, dan platform promosi, diagram ini digunakan untuk memodelkan entitas utama seperti *User*, *Product*, *Order*, dan *Transaction*, serta hubungan antar entitas yang kemudian

direalisasikan dalam tabel basis data atau objek dalam kode program (Yogi Swara dkk., 2023).

Secara praktis, *Class Diagram* digunakan untuk menstandarkan struktur data dan antarmuka antar modul, mendukung proses pemeliharaan melalui pendokumentasian tipe data dan relasi, serta menjadi dasar pembuatan repository atau *persistence layer* yang konsisten dengan model domain. Beberapa studi juga menunjukkan bahwa desain *Class Diagram* yang matang mempermudah implementasi pada berbagai framework seperti *Flutter* dan *Firebase* atau *PHP/MySQL*.

Tabel 2. 3 Class Diagram

No	Gambar	Nama	Keterangan
1		<i>Class</i>	Kelas pada struktur sistem.
2		<i>Interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3		<i>Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
4		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
5		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
6		<i>Depedency</i>	Relasi antar kelas dengan kebergantungan antar kelas.
7		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (wholepart).

Sumber: (Harlina, 2025)

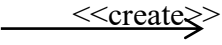
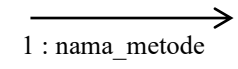
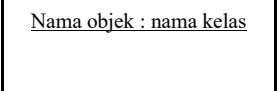
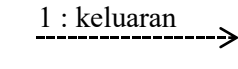
2.4.4. Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek atau komponen dalam urutan waktu, sehingga membantu menjelaskan bagaimana suatu skenario *use case* dijalankan secara dinamis. Diagram ini menunjukkan *lifeline* objek, pesan sinkron dan asinkron, serta aktor yang memulai proses. Dalam konteks aplikasi web atau mobile, diagram ini digunakan untuk memodelkan komunikasi antara antarmuka pengguna, *controller* atau layanan, dan penyimpanan data seperti basis data atau *API* eksternal. Contoh alur yang sering digambarkan mencakup autentikasi, proses *checkout*, serta sinkronisasi data pada platform seperti *Firebase/Firestore* (Stianingsih dkk., 2023).

Manfaat utama diagram ini adalah kemampuannya untuk memvalidasi alur eksekusi, mengidentifikasi kebutuhan *concurrency* atau *timeout handling*, serta menjadi cetak biru untuk pengujian integrasi dan pembuatan *mock* atau *stub* antar komponen. Banyak penelitian menggunakan *Sequence Diagram* bersama dengan *Activity Diagram* dan *Class Diagram* untuk memastikan bahwa struktur statis dan perilaku dinamis selaras sebelum proses implementasi.

Tabel 2. 4 *Sequence Diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Orang, proses, atau system lain yang berinteraksi dengan system informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang.
2		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.

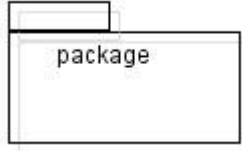
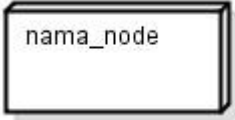
3		Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
4		Pesan tipe call	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
5		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
6		Objek	Menyatakan objek yang berinteraksi pesan.
7		Pesan tipe return	Menunjukkan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya.

Sumber: (M.Shalahuddin, 2022)

2.4.5. Deployment Diagram

Deployment diagram merupakan salah satu jenis diagram UML yang menggambarkan penempatan (*deployment*) komponen software pada perangkat keras (*hardware*), termasuk hubungan fisik antar node dan komunikasi protokol seperti HTTP atau TCP/IP. Diagram ini membantu memvisualisasikan arsitektur fisik sistem secara detail. *Deployment diagram* menunjukkan bagaimana komponen *software* dideploy ke dalam infrastruktur sistem, di mana sistem direpresentasikan oleh *node-node* berbentuk kubus 3D yang melambangkan perangkat keras seperti *server* atau *client*. Elemen kunci meliputi *node (hardware)*, *artifact (software yang dideploy)*, dan *communication path* (koneksi antar node) (Bibiola dkk., 2023).

Tabel 2. 5 *Deployment Diagram*

No	Gambar	Nama	Keterangan
1		<i>Package</i>	Package merupakan sebuah bungkus dari satu atau lebih.
2		<i>Node</i>	<i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang telah didefinisikan sebelumnya pada diagram komponen.
3		<i>Dependency</i>	Kertergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
4		<i>Link</i>	Relasi antar <i>node</i> .

Sumber: (M.Shalahuddin, 2022)

2.5. Sistem Informasi Desa (SID)

Berbagai penelitian mengungkapkan bahwa Sistem Informasi Desa (SID) adalah suatu kumpulan perangkat keras, perangkat lunak, prosedur, dan unsur manusia yang disusun secara sistematis untuk mengumpulkan, menyimpan, mengolah, dan menyajikan data serta informasi terkait pelaksanaan pemerintahan desa dan pembangunan berbasis perdesaan. Sistem ini berfungsi untuk mendukung proses pengambilan keputusan, penyediaan layanan publik, dan memperkuat partisipasi masyarakat. Dalam konteks implementasi *OpenSID*, SID dipahami sebagai "sekumpulan instrumen dan proses yang menggunakan data serta informasi untuk mendukung pengelolaan sumber daya berbasis komunitas di tingkat desa," yang terkait langsung dengan ketentuan dalam Undang-Undang Desa Pasal 86 (Hamza dkk., 2022). Di sisi lain, pendekatan yang lebih teknis menggambarkan

SID sebagai sistem pengolahan data berbasis komputer yang dikelola oleh pemerintah dan komunitas desa untuk menciptakan transparansi, akuntabilitas, serta partisipasi warga dalam proses pembuatan kebijakan.

2.5.1. Tujuan dan Fungsi SID

Hasil sintesis dari berbagai penelitian empiris mengungkapkan bahwa SID memiliki berbagai tujuan dan fungsi yang saling terkait. SID mendukung kegiatan administrasi dan manajemen pemerintahan desa dengan menyediakan basis data yang mencakup informasi penduduk, aset, keuangan, serta dokumen perencanaan, sehingga proses dokumentasi dan pelaporan menjadi lebih terorganisir (Romadhoni & Hwihanus, 2024). Selain itu, SID berperan dalam meningkatkan transparansi dan akuntabilitas karena masyarakat dapat mengakses data dan informasi terkait pembangunan desa melalui portal atau situs web, termasuk kebijakan, anggaran, dan realisasi program (Prawiro dkk., 2023).

Di sisi lain, SID memungkinkan penyediaan layanan publik digital yang lebih cepat dan efisien dengan mengurangi ketergantungan pada prosedur manual dalam proses administrasi maupun permintaan layanan. Data yang terintegrasi juga memudahkan pemerintah desa dalam menyusun perencanaan berbasis bukti serta melaksanakan kegiatan monitoring dan evaluasi dengan cara yang lebih sistematis (Hartono dkk., 2025). Selain itu, SID juga berfungsi sebagai sarana komunikasi dan partisipasi warga melalui distribusi informasi, penampungan aspirasi, dan keterlibatan masyarakat dalam kegiatan pembangunan desa (Sihombing dkk., 2025). Penelitian tentang penerapan *OpenSID*, *Siskeudes/SIMDES*, serta

pengembangan sistem berbasis *design science* menegaskan bahwa SID merupakan elemen kunci dalam strategi Desa Cerdas dan *e-government* desa.

2.5.2. Manfaat SID bagi Pemerintah Desa dan Masyarakat

Manfaat SID yang teridentifikasi dalam berbagai penelitian menunjukkan dampak positif yang signifikan bagi pemerintah desa dan masyarakatnya. Bagi pemerintah desa, SID meningkatkan efisiensi administrasi dan kinerja pemerintahan, serta mempermudah proses pelaporan kepada pemerintah kabupaten/kota. Integrasi SID dengan sistem keuangan desa seperti *Siskeudes/SIMDES* menjadikan pengelolaan alokasi dana desa lebih sesuai dengan regulasi dan lebih *akuntabel* (Erida dkk., 2023). Studi lapangan di Kabupaten Simalungun menunjukkan bahwa penerapan *Siskeudes* yang terintegrasi dengan SID dapat meningkatkan kualitas layanan dan memperkuat kepatuhan terhadap regulasi keuangan desa.

Bagi masyarakat, SID memberikan akses informasi yang lebih cepat dan luas, mempermudah layanan publik, serta mendorong partisipasi warga dalam perencanaan dan pengawasan pembangunan desa. Hasil dari program pelatihan dan sosialisasi *OpenSID* menunjukkan bahwa pemahaman warga tentang fungsi SID meningkat, dan warga menjadi lebih mudah dalam mengakses layanan administratif. Dari sudut pandang pembangunan, SID membantu pemerintah desa merumuskan kebijakan berbasis data yang lebih akurat, sehingga alokasi sumber daya menjadi lebih efisien dan program pembangunan lebih berkelanjutan (Hartono dkk., 2025). Penelitian berbasis *design science* juga menunjukkan potensi SID dalam memperkuat pembangunan desa yang partisipatif dan terukur.

2.5.3. Regulasi dan Kebijakan Pemerintah tentang SID

Penerapan SID didasarkan pada kerangka hukum yang terkait dengan Undang-Undang Nomor 6 Tahun 2014 tentang Desa, khususnya pasal-pasal yang mengatur sistem informasi pembangunan desa dan pentingnya penggunaan platform digital untuk dokumentasi serta pelaporan pembangunan. Selain itu, berbagai peraturan teknis seperti Permendagri dan panduan *Siskeudes/SIMDES* menjadi acuan dalam pengembangan modul akuntansi SID untuk mendukung tata kelola keuangan desa sesuai dengan regulasi (Erida dkk., 2023). Penelitian tentang implementasi SID menekankan bahwa sinergi antara kebijakan pusat/daerah dan kapasitas desa adalah syarat penting untuk memastikan sistem dapat berjalan secara efektif.

2.5.4. Tantangan Implementasi SID di Desa

Berbagai penelitian mengungkapkan bahwa penerapan SID menghadapi sejumlah hambatan, baik yang berasal dari dalam desa maupun faktor eksternal. Salah satu tantangan utama adalah keterbatasan kemampuan sumber daya manusia, khususnya operator dan perangkat desa, yang masih membutuhkan pelatihan dan pendampingan intensif untuk mengelola SID secara efektif (Karimullah dkk., 2022)(Erida dkk., 2023). Selain itu, banyak desa menghadapi masalah kualitas dan kelengkapan data administrasi yang belum memadai, sehingga penyusunan basis data SID menjadi sulit.

Keterbatasan infrastruktur teknologi informasi, seperti masalah konektivitas dan akses internet, juga menjadi hambatan dalam pemanfaatan SID, sebagaimana terlihat dalam studi pengoptimalan website SID di Bojonegoro. Selain itu,

ketidaksesuaian fitur aplikasi dengan kondisi lokal menyebabkan rendahnya tingkat adopsi, yang mengharuskan adanya penyesuaian sistem dan pendekatan partisipatif, seperti yang ditunjukkan pada studi Desa Cukilan dan Desa Rempek (Prawiro dkk., 2023). Hambatan lainnya mencakup kurangnya dukungan kebijakan dan pendanaan dari pemerintah daerah, serta rendahnya partisipasi masyarakat yang disebabkan oleh kurangnya pemahaman tentang manfaat SID.

2.6. Desa Digital dan Media Informasi Desa

Desa digital merujuk pada pemanfaatan teknologi informasi dan komunikasi di tingkat desa untuk meningkatkan efisiensi layanan publik, memperkuat perekonomian lokal, dan memperluas akses informasi bagi masyarakat. Konsep ini muncul dari kebutuhan untuk mendorong digitalisasi dalam administrasi dan pelayanan desa, sehingga tata kelola menjadi lebih transparan, responsif, dan partisipatif. Berbagai kajian implementasi dan kegiatan pendampingan menunjukkan bahwa transformasi menuju desa digital mencakup pembangunan infrastruktur informasi seperti situs web desa, peningkatan kapasitas sumber daya manusia perangkat desa, serta penggunaan platform digital untuk mengelola potensi ekonomi dan promosi desa (Maghfiroh dkk., 2023).

2.6.1. Pengertian Desa Digital

Secara operasional, desa digital diartikan sebagai desa yang mengintegrasikan teknologi informasi dalam penyelenggaraan layanan administrasi publik, peningkatan kualitas tata kelola pemerintahan desa, serta pengembangan kegiatan ekonomi lokal dan komunikasi sosial masyarakat. Penelitian tentang transformasi digital di desa menyoroti bahwa teknologi informasi memainkan peran

penting dalam mempercepat berbagai proses administratif dan mendukung penguatan ekonomi melalui pemanfaatan media sosial dan platform e-commerce (Nurhadryani dkk., 2025). Studi tentang digitalisasi Desa Pongkok menunjukkan bahwa penggunaan sistem informasi tidak hanya mempercepat layanan publik, tetapi juga memperluas jangkauan pemasaran produk UMKM melalui saluran digital. Meskipun demikian, tantangan umum yang dihadapi adalah rendahnya tingkat literasi digital serta keterbatasan keterampilan perangkat desa, sehingga dibutuhkan pelatihan dan pendampingan berkelanjutan agar transformasi digital dapat berjalan efektif (Nuraisiah dkk., 2023).

2.6.2. Peran *Website* sebagai Media Informasi Desa

Website desa berfungsi sebagai saluran resmi untuk menyampaikan informasi publik, profil desa, layanan administrasi, dan mempublikasikan berbagai potensi lokal. Penelitian dan pengalaman pendampingan menunjukkan bahwa website desa menjadi alat penting untuk menyediakan informasi secara cepat, menyimpan dan menampilkan berbagai layanan, serta mempromosikan potensi wisata dan ekonomi lokal. Studi pendampingan di Desa Hajimena dan Desa Keramas menunjukkan bahwa pelatihan pengelolaan website meningkatkan kemampuan perangkat desa dalam membuat dan memperbarui konten, termasuk layanan administrasi berbasis web dan profil desa, sehingga website dapat berfungsi secara maksimal sebagai media informasi (Eka Putra dkk., 2023).

Integrasi layanan tambahan seperti perpustakaan digital juga memperkaya fungsi website desa, menjadikannya repositori informasi edukatif yang mendukung peningkatan literasi masyarakat. Dengan demikian, website desa memiliki peran

strategis sebagai pusat informasi dan saluran pelayanan publik yang mudah diakses oleh seluruh warga.

2.6.3. Kendala Implementasi Sistem Informasi Desa

Implementasi sistem informasi di desa menghadapi berbagai hambatan, mulai dari terbatasnya kemampuan SDM desa dalam mengoperasikan atau mengembangkan sistem, ketidaklengkapan data administrasi, ketidaksesuaian aplikasi dengan karakteristik lokal, hingga kurangnya dukungan teknis dan kebijakan dari pemerintah daerah (Karimullah dkk., 2022). Namun, berbagai penelitian juga menunjukkan bahwa kesiapan implementasi dapat ditingkatkan melalui pelatihan dan pendampingan intensif, pemilihan platform yang tepat seperti *OpenSID*, serta penerapan arsitektur TI yang memadai untuk mendukung tata kelola elektronik desa.

2.6.4. Media Informasi dalam *Good Governance*

Untuk memperkuat pemanfaatan sistem informasi di desa, sejumlah langkah strategis dapat diterapkan berdasarkan temuan penelitian yang ada. Pendampingan jangka panjang melalui sosialisasi, pelatihan teknis, serta monitoring dan evaluasi perlu dilakukan secara konsisten sebagai upaya untuk meningkatkan kapasitas perangkat desa dan masyarakat (Firdaus dkk., 2022). Selain itu, pemilihan platform yang sesuai dengan kebutuhan lokal seperti *OpenSID* dan perancangan antarmuka yang mudah dipahami akan membantu meningkatkan penerimaan serta efektivitas penggunaannya. Proses pengumpulan dan verifikasi data administrasi desa juga perlu distandardisasi untuk memastikan kualitas informasi yang dihasilkan. Lebih jauh, penggunaan arsitektur TI dan metodologi

pengembangan seperti *AD/EDM* atau model *SDLC Waterfall* dapat memperkuat integrasi layanan, keamanan data, serta mengakomodasi kebutuhan pengembangan sistem di masa mendatang.

2.7. Penelitian Terdahulu

Penelitian mengenai sistem informasi desa berbasis web mengungkapkan adanya beberapa celah yang perlu diperbaiki untuk meningkatkan pemahaman dan implementasi sistem tersebut. Salah satunya adalah kurangnya perhatian terhadap tantangan implementasi yang dihadapi oleh desa-desa terpencil dengan infrastruktur yang terbatas, tingkat literasi yang rendah, dan tata kelola yang tidak optimal. Meskipun beberapa penelitian telah membahas konsep umum sistem informasi desa, penelitian yang lebih mendalam tentang konteks lokal yang kompleks, seperti yang dihadapi oleh Desa Bandar Kumbul, masih jarang ditemukan. Penelitian oleh (Muhammad & Dara Aisyah, 2024) menunjukkan bahwa sistem informasi berbasis web dapat meningkatkan kualitas penyampaian informasi yang akurat, namun tidak membahas bagaimana konteks lokal dapat memengaruhi penerapan sistem tersebut.

Tabel 2. 6 Peneliti Terdahulu

Judul	Nama Peneliti	Tahun	Hasil
Pengembangan Sistem Informasi Transparansi Desa Berbasis Web	Anjani & Wicaksana	2024	Website meningkatkan transparansi anggaran dan mempermudah masyarakat mengakses informasi desa secara mandiri.
Perancangan Sistem Informasi Desa Tomuan Holbung	Supiyandi , Muhammad Zen	2022	Sistem informasi desa berbasis web yang dirancang dengan metode <i>Waterfall</i>

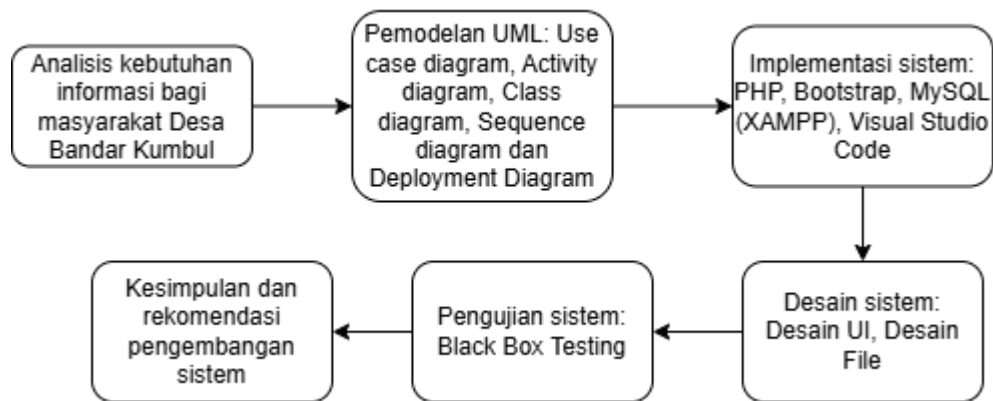
Judul	Nama Peneliti	Tahun	Hasil
Menggunakan Metode <i>Waterfall</i>			terbukti sesuai dengan kebutuhan pengelolaan informasi di Desa Tomuan Holbung, membantu transparansi, kemudahan akses informasi publik, dan efektivitas kerja perangkat desa.
Penerapan Sistem Informasi Desa menggunakan <i>OpenSID</i> Pada Desa Permata	Rampi Yusuf, Lillyan	2023	Penerapan <i>OpenSID</i> di Desa Permata terbukti berhasil dalam meningkatkan sumber informasi, memperkuat transparansi.
Pengembangan Sistem Informasi Desa Cukilan Menggunakan Pendekatan Design Science Research	A Prawiro & JJC Tambotoh	2023	DSR memberikan solusi yang lebih terstruktur dan sesuai kebutuhan nyata di desa Cukilan, meningkatkan kualitas pengelolaan informasi desa.
Pengembangan dan Implementasi Sistem Informasi Kelurahan Berbasis Website (Studi Kasus: Kantor Kelurahan Pulau Buluh, Kota Batam)	Ahmadi Irmansyah Lubis, Alena Uperiati	2023	<i>OpenSID</i> yang dikembangkan terbukti bermanfaat untuk meningkatkan dan memberikan kemudahan akses informasi kepada masyarakat setempat
Sistem Informasi Desa Berbasis Web Untuk Meningkatkan Keterlibatan Masyarakat	Kurniawati & Mursyidah	2023	Sistem informasi berbasis web yang dibangun berhasil meningkatkan keterlibatan masyarakat dalam proses perencanaan dan pengawasan pembangunan desa.
Perancangan dan Implementasi Sistem Informasi Desa	Nurhikmahyanti dkk	2024	Implementasi <i>OpenSID</i> pada desa X berhasil meningkatkan

Judul	Nama Peneliti	Tahun	Hasil
Menggunakan <i>OpenSID</i> di Desa X			transparansi data desa dan mempercepat pengelolaan administrasi desa.
Sistem Informasi Desa Berbasis Web untuk Optimalisasi Layanan Publik	Karina & Rahmat	2025	Penggunaan sistem informasi berbasis web meningkatkan kecepatan dan kemudahan dalam mengakses layanan publik bagi masyarakat.
Aplikasi Sistem Informasi Desa Berbasis Web dalam Meningkatkan Efisiensi Administrasi Pemerintahan Desa	Sucipto dkk	2024	Implementasi aplikasi web untuk administrasi desa mengurangi beban kerja perangkat desa dan meningkatkan efisiensi dalam pengelolaan data.
Pengembangan Sistem Informasi Desa untuk Penyediaan Data Kesejahteraan Sosial Berbasis Web	Wicaksono dkk	2025	Sistem informasi desa berbasis web berhasil mendukung penyediaan data kesejahteraan sosial yang lebih akurat dan mudah diakses oleh masyarakat dan pemerintah desa.

2.8. Kerangka Penelitian

Kerangka penelitian menggambarkan tahapan sistematis dalam pengembangan sistem berbasis teknologi informasi, dimulai dari analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian dan evaluasi. Setiap langkah bertujuan untuk memastikan bahwa sistem yang dikembangkan memenuhi kebutuhan pengguna dan berfungsi dengan optimal. Dengan menggunakan model *UML* untuk pemodelan, pengembangan dilakukan

menggunakan alat dan teknologi terbaru, serta diuji dengan metode yang tepat untuk memastikan kualitas dan efektivitas sistem.



Gambar 2. 6 Kerangka Penelitian

1. Analisis kebutuhan informasi bagi masyarakat Desa Bandar Kumbul

Pada tahap awal, dilakukan analisis terhadap kebutuhan informasi yang diperlukan oleh masyarakat di desa tersebut. Ini melibatkan pemahaman tentang masalah yang dihadapi masyarakat dan bagaimana informasi dapat membantu menyelesaikan masalah tersebut.

2. Pemodelan UML: *Use case diagram*, *Activity diagram*, *Class diagram*, *Sequence diagram* dan *Deployment Diagram*

Setelah kebutuhan teridentifikasi, langkah berikutnya adalah pemodelan sistem menggunakan *Unified Modeling Language* (UML).

3. Implementasi sistem: PHP, Bootstrap, MySQL (XAMPP), Visual Studio Code

Pada tahap ini, sistem yang telah dimodelkan akan diimplementasikan menggunakan teknologi tertentu seperti: PHP untuk pemrograman sisi server, Bootstrap untuk desain tampilan responsif (*front-end*), MySQL untuk pengelolaan database, XAMPP digunakan sebagai platform lokal

untuk mengembangkan aplikasi berbasis web, Visual Studio Code digunakan sebagai editor untuk menulis kode.

4. Desain sistem: Desain UI, Desain File

Pada bagian ini, fokusnya adalah mendesain antarmuka pengguna (UI) dan struktur file yang akan digunakan dalam sistem. Desain UI bertujuan untuk memastikan antarmuka yang mudah digunakan oleh pengguna.

5. Pengujian sistem: *Black Box Testing*

Setelah implementasi selesai, dilakukan pengujian sistem menggunakan metode *Black Box Testing*, yang berfokus pada pengujian fungsi sistem tanpa mengetahui struktur internalnya. Pengujian ini bertujuan untuk memastikan bahwa sistem bekerja sesuai dengan yang diinginkan.

6. Kesimpulan dan rekomendasi pengembangan sistem

Setelah pengujian, kesimpulan diambil tentang kinerja dan efektivitas sistem. Berdasarkan pengujian, diberikan rekomendasi untuk pengembangan lebih lanjut atau perbaikan sistem di masa depan.