

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian oleh Jessica Fransisca Tonapa dkk. dalam jurnal “Digitalisasi Pelayanan Administrasi dan Pengembangan Website Resmi Desa Benteng” bertujuan mengembangkan *website* desa sebagai *platform* informasi terpadu yang memuat profil desa, data kependudukan, layanan administrasi dan publik, program kerja, serta potensi desa seperti UMKM dan BUMDes melalui situs desabentengku.my.id. Metode yang digunakan meliputi pengumpulan dokumen desa, dokumentasi kegiatan, pembuatan *website*, serta pelatihan staf untuk mengurangi penggunaan kertas dan meningkatkan akses digital. *Website* ini dilengkapi fitur visi-misi, struktur organisasi, data kependudukan, sarana prasarana, dan layanan surat. Hasilnya, *website* mampu meningkatkan transparansi informasi, mempermudah pelayanan administrasi, serta mendukung promosi UMKM guna pemberdayaan masyarakat. [6].

Pada penelitian lain yang dilakukan oleh Zul Rachmat, Rudi Salam, dan Andi Supriadi dengan judul “Sistem Informasi Pelayanan Administrasi pada Desa Abbanuangne Kabupaten Soppeng”, membahas permasalahan pelayanan administrasi yang masih manual, sehingga pembuatan surat memakan waktu lama, rawan kesalahan pencatatan, dan sulit dalam pencarian arsip. Untuk mengatasinya, dirancang sistem informasi berbasis *web*

menggunakan metode *waterfall* dengan basis data penduduk agar pembuatan surat lebih cepat tanpa pengetikan ulang. Hasilnya, pelayanan menjadi lebih efisien, kesalahan input berkurang, dan arsip tersimpan secara digital sehingga mudah diakses. Berbeda dari penelitian sebelumnya, penelitian ini lebih menekankan pada peningkatan proses pelayanan surat secara langsung. Dengan demikian, penelitian Abbanuangnge memiliki fokus pada otomatisasi layanan administrasi harian agar lebih cepat dan tertata[1].

Penelitian lainnya yang dilakukan oleh Yuslinah Nurdiani, Adelina Ibrahim, Arisandy Ambarita, dan Saiful Do. Abdullah berjudul “Penerapan Sistem Informasi Desa berbasis E-Government pada Desa Pulau Maitara” berfokus pada pembangunan sistem *E-Government* untuk memperluas penyampaian informasi dan mempromosikan potensi desa. Pengembangan dilakukan menggunakan metode *Systems Development Life Cycle* (SDLC) mulai dari analisis kebutuhan hingga uji coba sistem. Hasilnya, sistem informasi yang dibangun membantu pemerintah desa dalam mengelola informasi secara lebih efektif, mempermudah masyarakat memperoleh layanan, serta mendukung transparansi dan meningkatkan citra pelayanan pemerintah desa. [3].

2.2 Landasan Teori

2.2.1 Pemerintahan Desa

Pemerintahan desa adalah penyelenggaraan urusan pemerintahan dan kepentingan masyarakat setempat oleh

pemerintah desa berdasarkan Undang-Undang Nomor 6 Tahun 2014 tentang Desa. Pemerintahan desa berwenang mengatur penataan desa, menetapkan peraturan desa, serta melaksanakan pembangunan secara partisipatif guna meningkatkan kesejahteraan masyarakat dan mewujudkan pembangunan yang berkelanjutan [7].

2.2.2 Website

Website merupakan kumpulan halaman yang saling terhubung dalam jaringan internet dan digunakan sebagai media penyampaian informasi. Informasi yang disajikan dapat berupa teks, gambar, maupun unsur multimedia lainnya yang disusun berdasarkan topik tertentu. Melalui *website*, pengguna dapat mengakses dan memperoleh informasi secara mudah dan sistematis menggunakan peramban (browser), sehingga penyampaian informasi menjadi lebih efektif dan terstruktur [8].

2.2.3 Visual Studio Code

Visual Studio Code merupakan salah satu *Integrated Development Environment* (IDE) yang populer dan efektif digunakan oleh pengembang perangkat lunak. VS Code menyediakan berbagai fitur pendukung seperti *code highlighting*, *auto-completion*, dan *integrated terminal* yang memudahkan penulisan serta pengelolaan kode program. Selain itu, IDE ini mendukung berbagai bahasa pemrograman seperti *JavaScript*, *PHP*, *Python*, dan *C++* melalui ekstensi. Dengan dukungan fitur tersebut,

Visual Studio Code dapat membantu proses pengembangan aplikasi maupun *website* menjadi lebih sistematis, efisien, dan terorganisir [9]



Gambar 2.1 Visual Studio Code

2.2.4 Laragon

Laragon merupakan lingkungan pengembangan (development environment) yang digunakan untuk menjalankan dan mengelola aplikasi *web* secara lokal. *Platform* ini bersifat *universal*, cepat, ringan, dan terisolasi sehingga mampu mendukung berbagai bahasa pemrograman seperti PHP, *Node.js*, *Python*, *Java*, *Go*, dan *Ruby* (Laragon, n.d.). Selain menyediakan paket server dan basis data yang siap digunakan, *Laragon* juga dilengkapi fitur *automatic virtual host* dan *pretty URLs* yang memudahkan konfigurasi domain lokal serta pengelolaan struktur aplikasi, sehingga proses pengembangan dapat dilakukan secara lebih efisien dan terstruktur [10]



Laragon

Gambar 2. 2 Laragon

2.2.5 Laravel

Laravel merupakan *framework* PHP bersifat *open source* yang dikembangkan oleh Taylor Otwell pada tahun 2011. Sejak pertama kali dirilis, *Laravel* terus mengalami perkembangan hingga menjadi salah satu *framework* yang paling banyak digunakan dalam pengembangan aplikasi berbasis *web*. Keunggulan *Laravel* terletak pada kelengkapan fitur yang dimilikinya seperti *Artisan*, *Migration*, dan *Blade*, yang mendukung proses pengembangan aplikasi secara lebih sistematis [11].



Gambar 2. 3 Laravel

2.2.6 MySQL

MySQL merupakan salah satu *Database Management System* (DBMS) yang menggunakan SQL (*Structured Query Language*) sebagai bahasa utama dalam pengelolaan basis data. Sistem ini dirancang untuk mendukung banyak pengguna (*multi-user*) dan mampu menangani berbagai proses secara bersamaan (*multi-threaded*), sehingga banyak digunakan dalam pengembangan aplikasi berbasis *web* maupun sistem informasi. MySQL bersifat *open source* dan didistribusikan di bawah lisensi GNU *General Public License* (GPL), yang memungkinkan pengguna

memanfaatkannya untuk keperluan pribadi maupun komersial tanpa harus membayar biaya lisensi. Karena kemudahan penggunaan, kinerja yang stabil, serta dukungan komunitas yang luas, MySQL menjadi salah satu DBMS yang populer di berbagai kalangan pengembang [12].



Gambar 2. 4 MySQL

2.2.7 Database

Basis data (database) merupakan sekumpulan data yang saling berhubungan dan disimpan secara terstruktur dalam suatu sistem sehingga dapat diakses, dikelola, dan diperbarui secara efisien. Data yang tersimpan di dalamnya diorganisasikan untuk mengurangi redundansi serta menjaga konsistensi dan keamanan informasi. Dalam pengelolaannya, basis data umumnya menggunakan SQL (Structured Query Language) sebagai bahasa standar untuk melakukan proses manipulasi data, seperti menambah, mengubah, menghapus, dan menampilkan informasi sesuai kebutuhan. Dengan struktur dan mekanisme pengelolaan tersebut, basis data berperan penting dalam mendukung operasional sistem informasi dan analisis data [13].

2.2.8 HTML

HTML (*Hypertext Markup Language*) merupakan bahasa markup yang digunakan untuk menyusun dan menampilkan struktur halaman *web*. HTML berfungsi sebagai kerangka dasar dalam pembuatan *website* dengan mengatur elemen-elemen seperti paragraf, *heading*, tautan, gambar, dan berbagai komponen lainnya agar dapat ditampilkan melalui *browser*. Dalam pengembangannya, HTML bekerja bersama CSS dan *JavaScript*, di mana HTML bertugas membentuk struktur halaman, CSS mengatur tampilan dan tata letak, sedangkan *JavaScript* menambahkan interaktivitas. Perlu dipahami bahwa HTML bukanlah bahasa pemrograman, melainkan bahasa *markup* yang menggunakan tag-tag terstruktur untuk mendefinisikan dan menampilkan konten pada halaman *web* [14].



Gambar 2. 5 HTML

2.2.9 CSS

CSS (*Cascading Style Sheet*) merupakan bahasa yang digunakan untuk mengatur tampilan dan gaya pada halaman *web*. CSS berfungsi untuk mengendalikan elemen visual seperti warna, jenis dan ukuran huruf, tata letak, jarak antar elemen, hingga responsivitas tampilan pada berbagai perangkat. Tanpa CSS,

halaman *web* tetap dapat ditampilkan karena struktur dasarnya dibangun oleh HTML, namun tampilannya akan terlihat sederhana dan kurang menarik. Dengan penerapan CSS, desain *website* dapat dibuat lebih rapi, konsisten, dan estetis sesuai kebutuhan pengguna maupun pengembang [15].



Gambar 2. 6 CSS

2.2.10 PHP

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman *web* yang berjalan di sisi server (*server-side scripting*) dan digunakan untuk membangun aplikasi *web* dinamis. Awalnya dikenal sebagai *Personal Home Page*, PHP bersifat *open source* sehingga dapat digunakan dan dikembangkan secara bebas. Bahasa ini memiliki sintaks yang dipengaruhi oleh C dan C++ serta berfungsi untuk memproses data, mengelola logika aplikasi, dan berinteraksi dengan basis data. *File* PHP memiliki ekstensi *.PHP* dan dijalankan melalui server *web* [16].



Gambar 2. 7 PHP

2.2.11 Java Script

JavaScript merupakan bahasa pemrograman tingkat tinggi yang bersifat dinamis dan berperan penting dalam pengembangan *web* modern. Pada awalnya, *JavaScript* dijalankan di sisi klien (client-side) melalui *browser* untuk berinteraksi dengan HTML dan CSS. Namun, seiring perkembangannya, *JavaScript* juga dapat digunakan di sisi server melalui *Node.js* sehingga mampu menangani proses *backend* seperti pengolahan data dan komunikasi dengan basis data. Dengan dukungan berbagai *library* dan *framework* populer seperti *React*, *Vue*, dan *Angular*, *JavaScript* menjadi salah satu bahasa pemrograman utama yang banyak digunakan baik pada pengembangan *frontend* maupun *backend* [17]



Gambar 2. 8 Java Script

2.2.12 Tailwind CSS

Tailwind CSS adalah *framework* CSS berbasis *utility-first* yang dirancang khusus untuk membangun antarmuka (interface) *web* dengan tingkat responsivitas (mobile-friendly) yang tinggi langsung di dalam *file* HTML. Keunggulan utama *framework* ini terletak pada efisiensi pengaturan tampilan lintas perangkat melalui pendekatan *mobile-first* dan penggunaan *breakpoint* bawaan yang

praktis. Hanya dengan menambahkan awalan (prefix) pada kelas utilitasnya, elemen antarmuka dapat otomatis beradaptasi dan berubah ukuran secara konsisten saat diakses melalui *smartphone*, tablet, maupun monitor komputer tanpa perlu lagi menulis baris kode *query* kustom yang rumit di *file* CSS eksternal[18].



Gambar 2. 9 Tailwind

2.2.13 PHPmyadmin

PHPMyAdmin merupakan aplikasi berbasis *web* yang bersifat *open source* dan digunakan untuk mengelola basis data MySQL maupun MariaDB. Dikembangkan menggunakan bahasa PHP, *phpMyAdmin* menyediakan antarmuka grafis yang memudahkan pengguna dalam melakukan administrasi basis data, seperti membuat *database* dan tabel, mengelola data, mengatur relasi, serta menjalankan perintah SQL. Dengan fitur yang lengkap dan penggunaan yang relatif mudah, *PHPMyAdmin* menjadi salah satu *tools* yang banyak dimanfaatkan dalam pengembangan dan pengelolaan aplikasi berbasis *web* [19]

2.2.14 UML

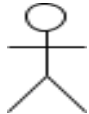
Unified Modeling Language (UML) merupakan bahasa pemodelan visual yang digunakan untuk merancang dan

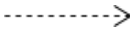
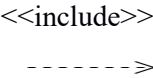
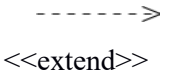
menggambarkan sistem perangkat lunak, khususnya yang berbasis konsep berorientasi objek. UML menyediakan berbagai diagram, seperti diagram *use case*, *class*, dan *activity*, yang berfungsi untuk menjelaskan kebutuhan sistem, struktur data, serta alur proses secara terstruktur. Dengan notasi dan simbol yang telah distandarisasi, UML membantu pengembang dan analis dalam memahami, merencanakan, serta mendokumentasikan sistem sebelum tahap implementasi [20].

Dalam praktik pengembangan sistem, beberapa diagram UML yang paling umum digunakan antara lain:

1. *Use Case Diagram* merupakan salah satu diagram dalam UML yang digunakan untuk menggambarkan fungsi-fungsi sistem serta interaksi antara aktor dan sistem yang dirancang. *Diagram* ini memperlihatkan layanan yang disediakan sistem dan pihak-pihak yang memiliki hak akses terhadap layanan tersebut. Aktor dapat berupa pengguna, proses, atau sistem lain di luar sistem yang dikembangkan, sedangkan *use case* merepresentasikan fungsionalitas yang disediakan oleh sistem [21].

Tabel 2. 1 *Use Case Diagram*

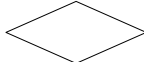
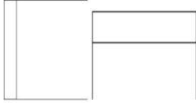
No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Orang, proses atau sistem yang berinteraksi dengan sisitem lain

No	Gambar	Nama	Keterangan
2.		<i>Dependency</i>	Hubungan perubahan yang terjadi pada satu elemen secara independen.
3.		<i>Generalization</i>	Hubungan generalisasi (umum-khusus) antar dua buah <i>use case</i>
4.		<i>Include</i>	Relasi <i>use case</i> tambah ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri
5.		<i>Extend</i>	Relasi tambahan yang menunjukkan <i>use case</i> lain memerlukan <i>use case</i> . Syarat untuk dijalankan atau agar fungsinya dapat berjalan dengan baik
6.		<i>Association</i>	Komunikasi antar aktor dan <i>use case</i> yang

No	Gambar	Nama	Keterangan
			berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
7.		<i>use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem
8.		<i>Note</i>	Elemen fisik yang ada saat aplikasi berjalan dan merepresentasikan sumber daya komputasi.

2. *Activity Diagram* merupakan diagram yang digunakan untuk memodelkan alur aktivitas atau proses yang terjadi dalam suatu sistem. *Diagram* ini menggambarkan urutan langkah secara sistematis dari awal hingga akhir proses, sehingga memperjelas bagaimana suatu aktivitas dijalankan. *Activity diagram* menampilkan alur kerja yang rinci, termasuk urutan menu maupun proses bisnis yang berlangsung di dalam sistem [22].

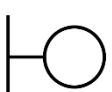
Tabel 2. 2 *Activity Diagram*

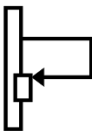
No	Gambar	Nama	Keterangan
1.		<i>Initial Node</i>	Bagaimana objek diawali, hanya ada satu initial dalam satu diagram.
2.		<i>Activity</i>	Memperlihatkan bagaimana kelas saling berinteraksi
3.		<i>Decision</i>	Asosiasi percabangan jika ada pilihan yang lebih dari satu
4.		<i>Fork Node</i>	Satu aliran pada tahap yang berubah menjadi beberapa aliran
5.		<i>Control Flow</i>	Menunjukkan urutan aliran aktivitas.
6.		<i>Partition</i>	Simbol yang membatasi aktivitas antar orang, organisasi, sistem
7.		<i>Activity Final Node</i>	Menjelaskan bagaimana suatu objek berakhir; dalam satu

No	Gambar	Nama	Keterangan
			diagram harus terdapat minimal satu <i>final state</i> .

3. *Sequence Diagram* merupakan merupakan *diagram* yang digunakan untuk menggambarkan perilaku objek dalam sebuah *use case* dengan mendeskripsikan alur proses melalui pesan-pesan yang dikirim antar objek. Oleh karena itu, untuk membuat *diagram sequence*, perlu dipahami objek-objek yang terlibat dalam *use case* beserta metode yang dimiliki oleh masing-masing objek[23].

Tabel 2. 3 *Sequence Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Entity Class</i>	Menjelaskan gambaran awal dari suatu sistem yang digunakan sebagai dasar dalam merancang serta menyusun struktur basis data.
2.		<i>Boundary Class</i>	bagian luar atau batas sistem yang berfungsi sebagai media interaksi

No	Gambar	Nama	Keterangan
			antara sistem dengan pengguna atau dengan sistem lainnya.
3.		<i>Control Class</i>	Penghubung antara <i>boundary</i> dengan <i>table</i> .
4.		<i>Recursive</i>	proses pengiriman pesan yang dilakukan oleh suatu objek kepada dirinya sendiri.
5.		<i>Activation</i>	Menghubungkan proses yang menunjukkan durasi aktivasi suatu proses.
6.		<i>Life Line</i>	Menghubungkan setiap objek yang memiliki <i>activation</i> dalam suatu proses.
7.		<i>Message</i>	Berperan sebagai pengirim pesan ke

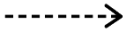
No	Gambar	Nama	Keterangan
			masing-masing <i>class</i> .

4. *Class Diagram* adalah *diagram* statis dalam UML yang memodelkan kelas, atribut, metode, dan hubungan antar objek. *Diagram* ini membantu merancang sistem atau aplikasi dengan menampilkan struktur data, relasi antar tabel atau objek, serta bagaimana data disimpan dan digunakan dalam *database* secara jelas [24].

Tabel 2. 4 *Class Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Class</i>	kumpulan objek yang memiliki atribut dan operasi yang sama sehingga dapat dikelompokkan dalam satu <i>class</i> .
2.		<i>Nary Association</i>	sebagai upaya untuk menghindari asosiasi yang melibatkan lebih dari dua objek.

No	Gambar	Nama	Keterangan
3.		<i>Collaboration</i>	Menunjukkan rangkaian aksi yang dilakukan oleh sistem yang menghasilkan suatu keluaran atau hasil bagi actor
4.		<i>Generalization</i>	Menunjukkan hubungan di mana objek turunan (<i>descendent</i>) memiliki perilaku dan struktur data yang diwariskan dari objek induk (<i>ancestor</i>).
5.		<i>Realization</i>	operasi atau tindakan yang dijalankan oleh suatu objek dalam sistem..

No	Gambar	Nama	Keterangan
6.		<i>Dependency</i>	Hubungan yang menunjukkan bahwa perubahan pada satu elemen dapat mempengaruhi elemen lain.