

BAB II

TINJAUAN PUSTAKA

2.1. Teori Terkait

Penelitian yang dilakukan oleh Bachtiar et al. (2022) membuktikan bahwa penerapan teknologi informasi melalui sistem *E-RT* mampu meningkatkan efisiensi dan transparansi pelayanan publik di tingkat masyarakat. Sistem ini dirancang berbasis web dan mobile agar dapat diakses kapan saja dan di mana saja oleh warga, memudahkan interaksi serta pelaporan secara real-time[1].

Penelitian yang menunjukkan bahwa sistem informasi digital berbasis Android mampu meningkatkan *efisiensi* dan akurasi dalam pengelolaan keuangan warga. Peryanto et al. (2025), misalnya, mengembangkan aplikasi sistem keuangan untuk RT 05 Karang Sari, yang memudahkan pencatatan kas masuk dan keluar, memberikan notifikasi pembayaran, serta menyajikan laporan keuangan bulanan secara otomatis. Sistem ini juga membantu pengurus RT dalam monitoring iuran dan menyederhanakan proses administrasi yang sebelumnya dilakukan secara manual dan rawan kesalahan[4].

Dalam penelitian (Jannah et al., 2024, hlm. 115–117) teknologi informasi dan komunikasi (TIK) memainkan peran penting dalam mempercepat proses pelayanan publik, memperluas akses informasi, serta meningkatkan efektivitas organisasi lokal dalam membangun komunikasi

antara pemerintah dan masyarakat. Penerapan teknologi dalam pelayanan publik juga meningkatkan partisipasi dan akuntabilitas. Hal ini membuktikan bahwa organisasi di tingkat desa, termasuk lembaga seperti RT, dapat memanfaatkan teknologi sebagai alat untuk peningkatan kinerja pelayanan masyarakat[2].

Penelitian terdahulu penggunaan teknologi dalam pengelolaan data dan sistem pembayaran *elektronik* menjadi aspek penting dalam meningkatkan efisiensi dan akurasi. Dalam konteks sistem informasi kas *RT*, integrasi antara pengelolaan data kas dan pembayaran non-tunai tidak hanya menyederhanakan proses pencatatan keuangan, tetapi juga memperbaiki transparansi dan kemudahan akses informasi bagi warga. Sistem manual yang masih digunakan pada pengelolaan kas *RT* saat ini sering menimbulkan masalah efisiensi dan akurasi, sehingga diperlukan sebuah solusi berbasis sistem informasi yang terintegrasi untuk mendukung pengelolaan keuangan secara lebih modern dan efektif[5].

2.2. Landasan Teori

2.2.1 Rukun Tetangga

Rukun Tetangga (*RT*) adalah organisasi masyarakat yang berada di tingkat paling kecil dalam suatu wilayah dan berfungsi sebagai sarana untuk membangun kerja sama, pelayanan, dan komunikasi antar warga. *RT* berperan dalam membantu administrasi kependudukan, menjaga ketertiban lingkungan, serta menjalankan

berbagai program kemasyarakatan yang ditetapkan oleh pemerintah atau masyarakat setempat[6].

2.2.2 Sistem Informasi

Menurut (Haposan P Simanungkalit J, 2012) dalam bukunya Konsep Sistem Informasi, sistem dapat diartikan sebagai suatu jaringan kerja prosedur yang saling terhubung dan terorganisasi untuk melaksanakan sebuah aktivitas atau mencapai tujuan tertentu. Fungsi utama sistem adalah untuk mengorganisasi proses, mengelola informasi, memecahkan masalah, meningkatkan efisiensi, dan mendukung pengambilan keputusan. Sebagai suatu entitas, sistem dirancang untuk memberikan solusi yang terstruktur dan efektif dalam berbagai konteks, baik manual maupun berbasis *teknologi*[7].

2.2.3 Website

Website dapat diartikan sebagai kumpulan halaman yang menyajikan berbagai informasi seperti: Misalnya data teks, gambar statis atau bergerak, animasi, suara, video atau kombinasi keduanya. Halaman-halaman ini dapat berupa konten statis atau dinamis yang terhubung satu sama lain melalui jaringan. Karena dapat menyampaikan informasi secara jelas dan terstruktur, *website* menjadi media informasi yang cepat, tepat dan akurat. Dengan menggabungkan unsur-unsur seperti teks, gambar, dan video, informasi dapat disajikan lebih detail dan lebih mudah dipahami pengguna[8].

2.2.4 Perangkat Lunak (*Software*)

Perangkat Lunak (*Software*) adalah sekumpulan instruksi *digital* atau program yang dirancang untuk menjalankan tugas-tugas tertentu pada perangkat komputer yang menghubungkan antara pengguna dan perangkat keras (*hardware*) memungkinkan pengguna untuk berinteraksi dengan perangkat komputer dan menjalankan tugas melalui penerjemahan instruksi ke bahasa mesin[9].

2.2.5 *Visual Studio Code*

Visual Studio Code (VS Code) adalah kode editor *open source* yang ringan namun kaya fitur, dikembangkan oleh Microsoft, dan mendukung berbagai bahasa pemrograman serta framework dengan kemampuan seperti *debugging*, manajemen Git, dan integrasi ekstensi untuk fleksibilitas[10].

2.2.6 *Framework Laravel*

Laravel merupakan *framework* pembuatan *website* yang mendukung sistem *MVC*. Model *MVC* sendiri merupakan metode membangun aplikasi dengan cara memisahkan data (*Model*), tampilan (*View*) dan jembatan pengolahan data (*Controller*). *Framework Laravel* memiliki kelebihan yaitu *sintaks* pemrogramannya menggunakan bahasa yang ekspresif sehingga memudahkan *programmer* untuk memahaminya. Pada tahun 2021, *Laravel* aktif digunakan oleh 724.681 *website* di seluruh dunia dan aktif digunakan

oleh 10.360 *website* di Indonesia. *Framework Laravel* memiliki fitur *Laravel mail* yang dapat mengirim *email* langsung dari *website*[11]..

2.2.7 Laragon

Laragon adalah platform pengembangan server lokal yang dirancang untuk menyediakan lingkungan pengembangan web yang sederhana, kuat, dan efisien. Dengan *Laragon*, pengembang dapat dengan mudah membuat server web lokal, mengelola basis data, dan menguji aplikasi web secara offline. *Laragon* juga mendukung berbagai bahasa pemrograman seperti PHP, Node.js, dan Python. Hal ini menjadikannya pilihan populer di kalangan pengembang untuk mempercepat proses pengembangan web, serta antarmuka intuitif yang memudahkan pengguna[12]

2.2.8 MySQL (*My Structured Query Language*)

My Structured Query Language (MySQL) adalah sebuah aplikasi sistem yang merupakan manajemen basis data relasional (*RDBMS*) yang menggunakan *Structured Query Language* (SQL) sebagai bahasa utamanya. Dikembangkan untuk mengelola, menyimpan, dan mengambil data dalam berbagai aplikasi, *MySQL* dikenal karena kecepatan, *stabilitas*, dan kemudahan penggunaannya. Sebagai perangkat lunak *open source*, *MySQL* sering digunakan dalam pengembangan aplikasi *website* dan sistem berbasis data seperti *Internet of Things* (IoT), menjadikannya salah satu pilihan populer di kalangan *programmer*[13].

2.2.9 Database

Database adalah sekumpulan data yang terorganisir dan tersimpan secara sistematis dalam suatu sistem komputer, sehingga dapat diakses dan dikelola menggunakan program untuk memperoleh data yang dibutuhkan selain itu database berfungsi untuk mencegah duplikasi serta inkonsistensi data yang mungkin terjadi apabila data dan informasi disimpan dalam jumlah yang banyak. Perangkat lunak yang digunakan untuk proses pengelolaan database dan menjalankan *Query* basis data disebut *Database Management System (DBMS)*[14].

2.2.10 HTML (*Hypertext Markup Language*)

HTML (*Hypertext Markup Language*) adalah bahasa *markup* yang digunakan untuk membuat dan merancang struktur halaman *web*. HTML hanya bertugas untuk menyusun tata letak dan menampilkan *konten* (teks, gambar, tabel, dll.) di *browser*[15].

2.2.11 CSS (*Cascading Style Sheets*)

CSS (*Cascading Style Sheets*) adalah bahasa yang digunakan untuk mengatur tampilan dokumen (*Content*) yang ditulis dengan bahasa *markup*. CSS paling sering digunakan untuk memformat halaman *web* yang ditulis dengan *HTML* dan *XHTML*, meskipun juga dapat diterapkan pada semua jenis dokumen *XML*, termasuk *SVG* dan *XUL*. Spesifikasi CSS ditetapkan oleh *World Wide Web Consortium (W3C)*.

CSS memungkinkan *developer* untuk mengatur *elemen* seperti warna, jenis font, tata letak, dan aspek tampilan lainnya.. Tujuan utama penggunaan CSS adalah untuk memisahkan isi dokumen (yang ditulis dengan *HTML* atau bahasa *markup* lainnya) dari tampilannya (yang diatur dengan *CSS*). Pemisahan ini tidak hanya meningkatkan *aksesibilitas konten*, tetapi juga memberikan lebih banyak *kontrol* atas desain dan mengurangi *kompleksitas* serta pengulangan dalam *struktur* dokumen[16].

2.2.12 Bootstrap

Bootstrap adalah kerangka kerja (*framework*) *front-end* yang membantu membangun website responsif dan aplikasi dengan komponen-komponen UI (*User Interface*) siap pakai (*open source*), seperti tombol dan navigasi, serta fitur grid system untuk memudahkan tata letak halaman dan desain responsif untuk meningkatkan konsistensi dan kemudahan akses pada berbagai perangkat[17].

2.2.13 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman sisi *server* yang digunakan untuk membuat *aplikasi web dinamis*. PHP bekerja di *backend* untuk memproses data sebelum ditampilkan di *browser* melalui *HTML*[18].

2.2.14 *Java Script*

Javascript adalah bahasa *pemrograman* tingkat tinggi yang digunakan untuk menambahkan interaktivitas, logika, dan *fungsionalitas* dinamis pada halaman *web*. *JavaScript* bekerja pada sisi *client-side* (*browser pengguna*), meskipun juga dapat digunakan di *server-side* dengan *platform* seperti *Node.js*[19].

2.2.15 *phpMyAdmin*

phpMyAdmin adalah aplikasi berbasis *web* yang digunakan untuk mengelola database *MySQL* atau *MariaDB*. *phpMyAdmin* menyediakan antarmuka grafis (*GUI*) yang memudahkan pengguna dalam mengelola database tanpa harus menggunakan perintah *SQL* secara manual meskipun demikian, di *phpMyAdmin* juga menyediakan untuk membuat *database* dengan bahasa *Query*[20].

2.2.16 *Midtrans*

Midtrans adalah perusahaan teknologi finansial (*fintech*) yang menyediakan layanan *payment gateway* untuk memfasilitasi pembayaran *online*. Midtrans membantu perusahaan dalam memproses transaksi pembayaran *online* dengan aman dan efisien[21]

2.2.17 *UML (Unified Modelling Language)*

Menurut sebuah penelitian, *Unified Modeling Language* (*UML*) adalah metode pemodelan visual yang digunakan untuk merancang sistem berbasis objek. *UML* didefinisikan sebagai bahasa standar yang berperan dalam visualisasi, perancangan, serta

pendokumentasian perangkat lunak. Keberadaan UML sangat penting dalam pengembangan sistem karena berfungsi sebagai penghubung antara pengembang dan pengguna[22].

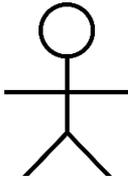
Dalam proses pengembangan sistem, terdapat beberapa diagram UML yang sering digunakan yaitu:

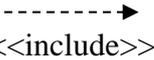
1. *Use case Diagram*

Usecase Diagram adalah salah satu jenis diagram UML (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor. Usecase memiliki simbol-simbol untuk penyampaian pesan, Tabel yang berisi simbol-simbol yang digunakan dalam Use Case Diagram dapat dilihat pada tabel 2.1

Simbol *Use case Diagram*

Tabel 2. 1. Use Case Diagram[23]

No	Simbol	Nama Simbol	Keterangan
1.		<i>Use Case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktif, yang dinyatakan dengan menggunakan kata kerja.
2.		<i>Actor/Aktor</i>	<i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi

No	Simbol	Nama Simbol	Keterangan
			aktif, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. aktor berinteraksi.
3.		<i>Association/ Assosiasi</i>	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
4.		<i>Generalization/ Generalisasi</i>	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk Mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
5.	 <<include>>	<i>Include/Uses</i>	Merupakan di dalam use case lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
6.	 <<extend>>	<i>Extend/Ekstensi</i>	perluasan dari <i>use case</i> lain jika syarat terpenuhi.

2. Activity Diagram

Diagram aktivitas adalah diagram yang digunakan untuk menggambarkan alur proses dalam sebuah sistem. *Activity* diagram merupakan pengembangan dari *usecase* diagram yang memperlihatkan langkah-langkah atau aktivitas dalam suatu proses. Tabel yang berisi Simbol-simbol yang digunakan dalam *Activity* Diagram dapat dilihat pada tabel 2.2.

Tabel 2. 2. Activity Diagram[23]

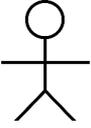
No	Simbol	Nama Simbol	Keterangan
1.		<i>Activity Action</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		<i>Activity Final Node</i>	bagaimana objek dibentuk dan dihancurkan.
3.		<i>Initial Node</i>	bagaimana objek dibentuk atau diawali.
4.		<i>Fork Node</i>	Satu aliran yang tahap tertentu berubah menjadi beberapa aliran.

3. Sequence Diagram

Sequence diagram adalah diagram yang menggambarkan tahapan atau interaksi antar objek dalam sebuah sistem berdasarkan

urutan waktu. Diagram ini menunjukkan bagaimana pesan dikirim dan diterima untuk menghasilkan alur yang sesuai dengan *usecase* diagram. Tabel yang berisi Simbol-simbol yang digunakan dalam *Sequence* Diagram dapat dilihat pada tabel 2.3.

Tabel 2. 3. Sequence Diagram[23]

No.	Simbol	Nama Simbol	Keterangan
1.		<i>Aktor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2.		<i>Entity Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.
3.		<i>Asynchronous Message</i>	Menggambarkan pengiriman pesan.
4.		<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.
5.		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem.
6.		<i>A Focus Of Control & A Life Line</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.

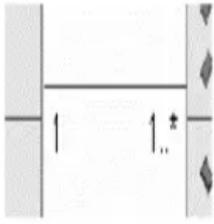
4. Class Diagram

Class diagram adalah diagram yang menampilkan struktur kelas dalam suatu sistem untuk memenuhi kebutuhan tertentu. Diagram ini merupakan penjabaran dari domain model, yang berfungsi sebagai abstraksi dari basis data atau bagian dari modul dalam

arsitektur MVC (*Model-View-Controller*). Tabel yang berisi Simbol-simbol yang digunakan dalam *Class Diagram* dapat dilihat pada tabel 2.4.

Tabel 2. 4. Class Diagram[23]

No.	Simbol	Nama Simbol	Keterangan
1.		Association	Relasi antar kelas dengan makna umum. Asosiasi ini menggambarkan bahwa satu kelas memiliki hubungan dengan kelas lain. Biasanya disertai dengan multiplicity untuk menunjukkan banyaknya objek yang saling terhubung.
2.		Direction association	Relasi antar kelas dengan makna bahwa satu kelas menggunakan atau mengenal kelas lain. Panah menunjukkan arah hubungan (navigasi), sehingga jelas kelas mana yang mengetahui kelas lainnya.
3.		Generalization	Relasi antar kelas dengan makna umum-khusus (inheritance), di mana superclass diturunkan menjadi subclass.

No.	Simbol	Nama Simbol	Keterangan
4.		Dependency	Relasi antar kelas dengan makna ketergantungan, yaitu satu kelas membutuhkan kelas lain secara lemah atau sementara, digambarkan dengan garis putus-putus.
5.		Aggregation	Relasi antar kelas dengan makna whole-part (keseluruhan-bagian), di mana suatu kelas terdiri dari kelas lain, tetapi bagian tetap ada meskipun keseluruhan dihapus.
6.		Multiplicity	Menggambarkan jumlah objek dari suatu kelas yang dapat berhubungan dengan objek dari kelas lain. Contohnya: 1 → Tepat satu objek. 0..* → Nol atau lebih objek. 1..* → Satu atau lebih objek. 0..1 → Nol atau satu objek.