

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Menurut Sucipto dkk., (2022) dalam penelitiannya yang berjudul Penerapan Sistem Informasi Profil Berbasis Web di Desa Banjarsari menguraikan bahwa penerapan sistem informasi berbasis web di Desa Bandarsari memiliki dampak positif yang cukup signifikan. Sistem informasi ini terbukti dapat mempermudah dalam pengolahan data penduduk desa serta meningkatkan efisiensi dalam pengelolaan data dan informasi desa. Masyarakat desa juga akan lebih mudah untuk mengakses informasi dengan cepat dan tepat.

Penelitian yang dilakukan oleh Setiawan dkk., (2024) dengan judul Rancang Bangun Sistem Informasi Manajemen Angka Kredit Tenaga Kesehatan (SIMAK NAKES) Pada Dinas Kesehatan Kota Surakarta Berbasis Web menguraikan bahwa penerapan sistem informasi manajemen angka kredit tenaga kesehatan berbasis web memiliki dampak yang cukup signifikan dalam memberikan kemudahan dalam proses pengelolaan, pemantauan, serta penilaian angka kredit bagi jabatan fungsional di bidang kesehatan. Sistem informasi ini juga meningkatkan efektivitas dan efisiensi dalam pelaksanaan penilaian sehingga proses administrasi dapat berjalan lebih cepat, akurat, dan transparan.

Penelitian dari Saefudin dkk., (2023) dengan berjudul Penerapan *Framework* Laravel Pada Sistem Informasi Posyandu Berbasis Website menguraikan bahwa pemanfaatan *Framework* Laravel dapat memungkinkan pengelolaan data menjadi lebih efisien. Sistem posyandu berbasis web memudahkan kader dan masyarakat dalam mengelola data anak, imunisasi, ibu hamil, serta peserta KB. Selain itu sistem ini juga dapat meningkatkan ketepatan dan keakuratan laporan seluruh kegiatan dalam posyandu.

Menurut Restaldo & Beeh (2022) dalam penelitiannya yang berjudul Penerapan *Framework* Laravel pada Sistem Informasi Arsip Dosen Fakultas

Keguruan dan Ilmu Pendidikan menguraikan bahwa penggunaan *Framework* Laravel dalam pengembangan sistem informasi, khususnya pada sistem arsip dosen di Fakultas Keguruan dan Ilmu Pendidikan Universitas Kristen Satya Wacana, memiliki potensi besar dalam meningkatkan efisiensi dan efektivitas manajemen data. Struktur *Model-View-Controller* (MVC) pada Laravel memungkinkan sistem yang dihasilkan menjadi lebih terorganisir, sehingga mampu memenuhi kebutuhan informasi akademik dan non-akademik dengan lebih efisien.

Pada penelitian membangun sebuah sistem informasi berbasis web untuk pelayanan masyarakat di Kelurahan Jantiharjo, Kabupaten Karanganyar. Sistem ini akan dirancang menggunakan PHP (*Hypertext Preprocessor*) sebagai bahasa pemrograman dengan *Framework* Laravel, MySQL sebagai database server, serta Visual Studio Code sebagai aplikasi code editor. Sistem informasi ini akan memiliki output profil dari Kelurahan Jantiharjo serta layanan dari Kelurahan Jantiharjo, seperti layanan surat keterangan kelahiran, kematian, perkawinan, perceraian, domisili, pindah datang, tidak mampu, izin usaha, dan surat pengantar pembuatan SKCK. Terdapat tiga jenis pengguna dalam sistem informasi ini, yaitu admin dan operator sebagai petugas kelurahan yang memiliki wewenang pengelolaan sistem, serta masyarakat sebagai pengguna yang dapat mengakses layanan secara daring. Melalui sistem informasi ini diharapkan dapat mempermudah pengelolaan data oleh petugas kelurahan serta memberikan kemudahan bagi masyarakat untuk mendapatkan pelayanan dari kelurahan secara cepat dan efisien.

2.2 Kerangka Pemikiran

Menurut Aulia (2024) menguraikan bahwa kerangka pemikiran merupakan landasan pemikiran logis dalam suatu penelitian yang menjadi pokok utama dalam menentukan suatu permasalahan dalam penelitian serta menjadi acuan terhadap jalannya suatu penelitian. Adapun kerangka pemikiran yang dijalankan dalam penelitian ini adalah:

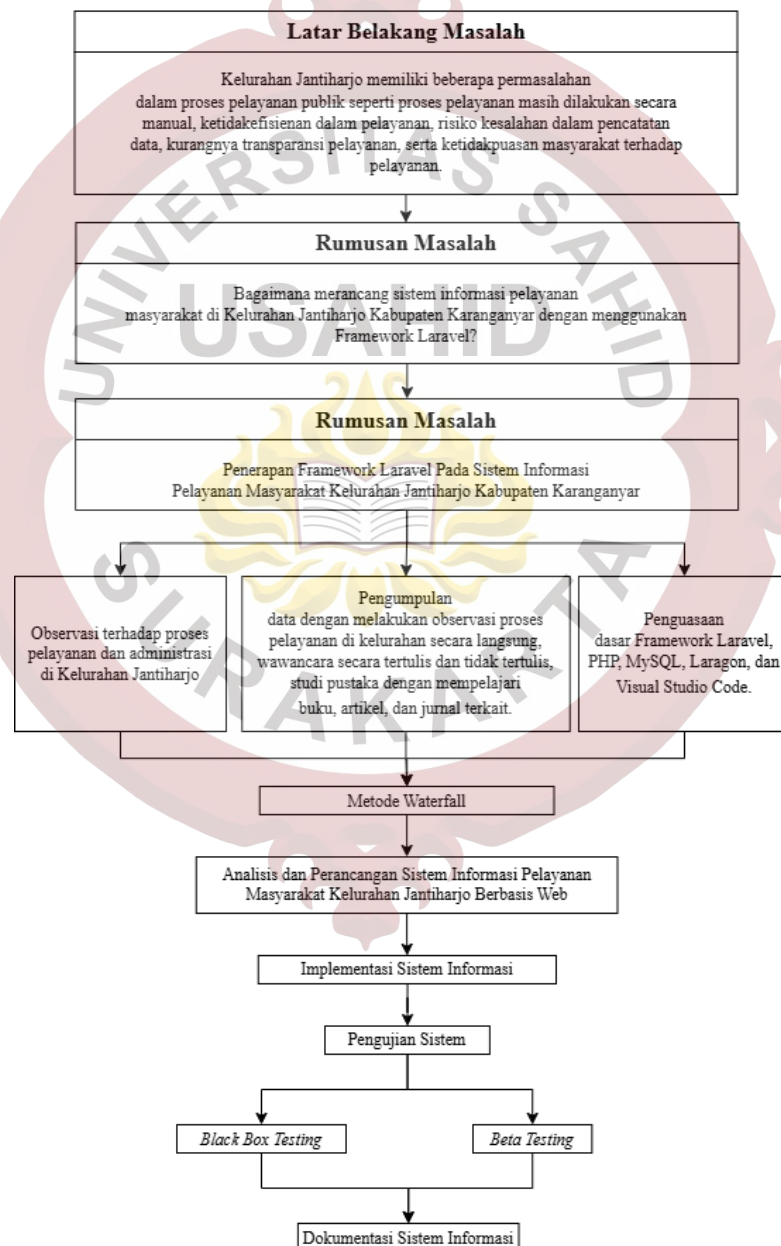
1. Latar belakang masalah
Tahap identifikasi permasalahan yang menjadi dasar penelitian.
2. Rumusan masalah
Tahap perumusan masalah berdasarkan latar belakang yang telah diidentifikasi.
3. Judul
Merupakan solusi yang diusulkan untuk menjawab rumusan masalah.
4. Observasi
Pengamatan langsung terhadap proses pelayanan masyarakat di Kelurahan Jantiharjo.
5. Pengumpulan data
Pengumpulan data melalui wawancara dan studi pustaka yang relevan dengan penelitian.
6. Penguasaan dasar
Mempelajari teknologi yang digunakan, seperti Laravel, PHP, MySQL, Laragon, dan Visual Studio Code.
7. Metode Waterfall
Merupakan metode pengembangan sistem yang digunakan dalam penelitian ini, yang terdiri dari tahapan analisis kebutuhan, perancangan sistem, implementasi, dan pengujian sistem.
8. Analisis Kebutuhan
Merupakan tahap identifikasi kebutuhan sistem berdasarkan hasil observasi dan pengumpulan data.
9. Perancangan Sistem
Merupakan tahap perancangan UML, basis data, dan antarmuka sistem sebagai acuan pengembangan.
10. Implementasi sistem informasi
Mengembangkan sistem berdasarkan hasil perancangan menggunakan Laravel.
11. Pengujian sistem

Merupakan tahap pengujian sistem menggunakan metode Black Box Testing dan Beta Testing untuk memastikan sistem berjalan sesuai kebutuhan.

12. Dokumentasi

Mendokumentasikan seluruh proses dan hasil penelitian ke dalam laporan tugas akhir.

Kerangka berpikir dapat disajikan pada Gambar 2.1.



Gambar 2. 1 Kerangka Pemikiran

2.3 Landasan Teori Pendukung

2.3.1 Pengertian Kelurahan

Kelurahan merupakan salah satu subdivisi administrasi terkecil dari satuan pemerintah yang keberadaannya diakui dalam undang-undang Nomor 23 Tahun 2014 tentang pemerintahan daerah dan peraturan pemerintah Nomor 37 tahun 2005 tentang kelurahan (Ritonga, 2024). Kelurahan merupakan bagian dari wilayah administratif yang memiliki batas-batas territorial yang memiliki wewenang untuk mengatur, mengelola serta mengurus kepentingan masyarakat setempat guna memenuhi kebutuhan dan meningkatkan kesejahteraan masyarakat setempat (Arman dkk., 2024).

2.3.2 Sistem Informasi

Sistem informasi merupakan salah satu kemajuan teknologi yang dapat meningkatkan efisiensi dalam suatu pekerjaan. Sistem informasi berperan sebagai alat untuk mempercepat dan mempermudah penggunaan serta pemrosesan data dalam suatu organisasi, institusi, atau badan usaha. Adanya sistem informasi ini dapat menjadikan pengambilan suatu keputusan menjadi lebih obyektif (Melyani dkk., 2023).

Menurut Arribe & Ryandi (2023) menguraikan bahwa sistem informasi merupakan suatu sistem yang mengintegrasikan antara manusia dan teknologi di dalam sebuah perusahaan atau organisasi. Sistem ini menyediakan informasi yang diperlukan bagi berbagai pihak serta menjadi penghubung antara kebutuhan dan aktivitas transaksi. Sistem informasi juga dapat berfungsi untuk membangun, mengelola, dan mengatur jaringan komunikasi.

2.3.3 Pelayanan Masyarakat

Menurut Primanto & Puspitasari (2024), dalam perkembangan teknologi penggunaan *website* sebagai media informasi dan administrasi pelayanan publik sangatlah penting. Tujuan utamanya adalah untuk mempermudah akses masyarakat terhadap informasi dan layanan administrasi. Adanya *website* masyarakat dapat dengan cepat dan mudah

mendapatkan informasi yang diperlukan tanpa harus datang langsung ke kantor kelurahan. Sistem informasi berbasis web diharapkan mampu mengurangi masalah dalam penyampaian dan pertukaran informasi, memudahkan masyarakat dalam mengakses informasi yang relevan dengan lebih efisien, serta meningkatkan transparansi dan akuntabilitas dalam pengelolaan layanan publik.

Lewat penyediaan layanan publik, pemerintah harus menyediakan layanan yang berkualitas agar masyarakat merasa puas dengan layanan yang telah diterima. Sistem pelayanan dibedakan menjadi dua jenis, yaitu sistem pelayanan online dan sistem pelayanan offline. Sistem pelayanan online merupakan sistem berbasis teknologi yang dilakukan melalui sebuah aplikasi, sedangkan sistem pelayanan offline adalah sistem manual di mana masyarakat harus datang ke kelurahan terlebih dahulu untuk mendapatkan layanan administrasi (Widuri dkk., 2024).

2.3.4 Framework Laravel

Laravel banyak digunakan dalam pengembangan aplikasi *website* karena memiliki kemudahan dalam pengelolaan, tingkat keamanan yang tinggi, serta dukungan terhadap arsitektur MVC (*Model-View-Controller*) yang memisahkan logika bisnis, tampilan, dan alur data dalam aplikasi. Laravel menawarkan fitur Eloquent ORM dan Blade Templating Engine yang membuat pengelolaan data menjadi lebih praktis dan tampilan database menjadi lebih dinamis (Sinlae dkk., 2024).

Model-View-Controller (MVC) merupakan metode pengembangan aplikasi yang membagi sistem ke dalam tiga komponen utama, yaitu *Model*, *View*, dan *Controller*. Komponen *Model* bertugas mengelola data serta berinteraksi langsung dengan basis data, *View* berfungsi menampilkan informasi kepada pengguna dalam bentuk antarmuka, sedangkan *Controller* berperan sebagai penghubung antara *Model* dan *View* dengan memproses permintaan pengguna serta menentukan tindakan yang harus dilakukan oleh sistem. Penerapan arsitektur MVC memungkinkan pengembangan aplikasi

menjadi lebih terstruktur, mudah dipelihara, serta meningkatkan efisiensi dan kualitas perangkat lunak yang dihasilkan (Restaldo & Beeh, 2022).

2.3.5 PHP

Menurut Zulfa & Wanda (2023), PHP atau *Hypertext Preprocessor* merupakan bahasa pemrograman yang sering digunakan untuk mengembangkan situs web dinamis. Pada proses pembuatan *website*, PHP berperan dalam pengolahan data yang dikirimkan oleh pengguna, memungkinkan adanya interaksi yang lebih kompleks antara server dan pengguna. PHP memiliki peran penting dalam pengembangan sistem informasi berbasis web, karena mendukung pengelolaan data dan penyampaian informasi kepada pengguna secara efisien.

2.3.6 MySQL

MySQL merupakan sistem manajemen basis data relasional (*Relational Database Management System/RDBMS*) yang menggunakan bahasa SQL (*Structured Query Language*) untuk mengelola, menyimpan, dan memanipulasi data. MySQL memiliki kemampuan untuk menangani banyak pengguna (*multi-user*), mendukung pengolahan data secara cepat, serta banyak digunakan dalam pengembangan aplikasi berbasis web karena kemudahannya integrasinya dengan bahasa pemrograman PHP dan framework Laravel (Ahmadar dkk., 2021). Pada penelitian ini, MySQL digunakan sebagai basis data untuk menyimpan dan mengelola data pelayanan masyarakat, seperti data pengguna, pengajuan surat, pengaduan masyarakat, dan informasi lainnya yang terdapat pada sistem informasi pelayanan masyarakat Kelurahan Jantiharjo.

2.3.7 Laragon

Laragon merupakan aplikasi pengembangan perangkat lunak serbaguna dan fleksibel yang dirancang untuk mendukung berbagai jenis bahasa pemrograman seperti PHP, Node.js, Python, Java, Go dan lain sebagainya. Laragon memiliki fokus pada kinerja optimal dengan mempertimbangkan aspek stabilitas yang tinggi, kesederhanaan dalam penggunaannya, fleksibilitas untuk memenuhi berbagai kebutuhan

pengguna, serta memberikan kebebasan yang luas kepada pengguna (Zulfa, 2024).

2.3.8 Visual Studio Code

Visual Studio Code merupakan sebuah aplikasi yang dirancang untuk pengembangan perangkat lunak dengan desain antarmuka yang sederhana serta mudah dipahami. Visual Studi Code memiliki dukungan bahasa pemrograman bawaan seperti JavaScript, Script, dan Node.js, serta berbagai bahasa pemrograman lain seperti C++, C#, Python, dan PHP. Kombinasi antara kemudahan pengguna, fleksibilitas, serta fungsionalitas membuat aplikasi Visual Studio Code menjadi pilihan yang sangat efektif dalam pengembangan perangkat lunak bagi programmer pemula maupun profesional (Mentari dkk., 2024).

2.3.9 Unified Modeling Language (UML)

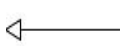
Unified Modeling Language (UML) merupakan suatu bahasa yang divisualisasikan dalam bentuk gambar atau grafik. UML terdiri dari diagram atau bagan yang menggambarkan bagaimana suatu sistem berfungsi, dengan fokus pada permasalahan yang dihadapi dan solusi yang dapat diterapkan. Diagram dan bagan ini dirancang untuk memfasilitasi pemahaman secara visual terhadap struktur, komponen, dan proses di dalam sistem, serta memberikan panduan yang jelas dalam pengembangan sistem yang efektif dan efisien (Narulita dkk., 2024).

a. Use Case Diagram

Penelitian yang berjudul *Sistem Informasi Pengelolaan Surat Masuk Dan Surat Keluar Pada Kantor Sekretariat Daerah Kabupaten Sumba Barat Daya* menguraikan bahwa *use case diagram* merupakan gambaran representasi visual mengenai bagaimana perilaku sistem yang akan dikembangkan dalam suatu proyek. *Use case diagram* akan mengidentifikasi siapa saja pengguna yang dapat mengakses serta menggunakan semua fitur yang tersedia dalam suatu sistem. Hal ini akan membantu memberikan gambaran yang jelas mengenai interaksi antara pengguna dengan sistem yang sedang dirancang (Gole dkk., 2024). Pada *use*

case diagram terdapat sekumpulan langkah-langkah yang dilakukan oleh aktor dalam berinteraksi dengan sistem (Ibrahim, 2024). Selain itu, *use case diagram* juga berfungsi untuk menggambarkan batasan sistem (*system boundary*) serta kebutuhan fungsional yang harus dipenuhi oleh sistem yang dikembangkan. Dengan adanya *use case diagram*, pengembang dapat lebih mudah memahami kebutuhan pengguna dan mengidentifikasi fitur-fitur yang diperlukan dalam sistem. Diagram ini juga menjadi acuan dalam proses perancangan dan pengembangan sistem sehingga setiap fungsi yang dirancang dapat sesuai dengan kebutuhan pengguna. Simbol-simbol dalam *use case diagram* dapat disajikan pada Tabel 2.1.

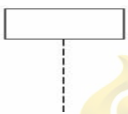
Tabel 2.1 Simbol *use case diagram* (Malius & Dani, 2021)

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	<i>Actor</i> menunjukkan peran pengguna yang berinteraksi dengan sistem untuk menjalankan fungsi tertentu.
2		<i>Generalization</i>	<i>Generalization</i> menunjukkan hubungan pewarisan antara elemen induk dan elemen turunan, sehingga elemen turunan mewarisi karakteristik dari elemen induknya.
3		<i>Include</i>	<i>Include</i> menunjukkan bahwa suatu use case selalu menggunakan use case lain sebagai bagian dari proses yang dijalankan.
4		<i>Extend</i>	<i>Extend</i> menunjukkan bahwa suatu use case dapat menambahkan fungsi atau proses pada use case lain dalam kondisi tertentu.
5		<i>Association</i>	<i>Association</i> menunjukkan hubungan atau interaksi antara aktor dengan use case dalam sistem..
6		<i>Use Case</i>	<i>Use Case</i> menunjukkan deskripsi dari urutan aksi-aksi yang ditampilkan sistem.

b. *Sequence Diagram*

Menurut Aman & Suroso (2021), *sequence diagram* umumnya digunakan untuk mendeskripsikan interaksi antara objek di dalam sebuah sistem. Diagram ini memperlihatkan jalannya setiap proses dari awal hingga akhir beserta langkah-langkah yang ada di dalam setiap proses. Penggunaan *sequence diagram* dapat mempermudah dalam mengenali alur kerja serta memastikan bahwa setiap langkah yang diperlukan terhubung dengan baik. *Sequence diagram* memberikan gambaran yang jelas mengenai urutan komunikasi antarobjek dalam sistem, sehingga dapat membantu pengembang dalam memahami dan merancang proses sistem secara lebih terstruktur. Simbol-simbol dalam *sequence diagram* dapat disajikan pada Tabel 2.2.

Tabel 2.2 Simbol *sequence diagram* (Malius & Dani, 2021)

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	<i>Lifeline</i> menunjukkan objek atau aktor yang terlibat dalam proses interaksi pada <i>sequence diagram</i> .
2		<i>Message</i>	<i>Message</i> menunjukkan komunikasi atau pengiriman pesan antar objek selama proses interaksi berlangsung.
3		<i>Message</i>	<i>Message</i> menunjukkan komunikasi atau pengiriman pesan antar objek selama proses interaksi berlangsung.
4		<i>Actor</i>	<i>Actor</i> menunjukkan pengguna yang berinteraksi dengan sistem untuk menjalankan fungsi tertentu.

c. *Activity Diagram*

Menurut Halim dan Somya (2023), *Activity Diagram* digunakan untuk menggambarkan alur aktivitas yang terjadi dalam suatu sistem secara visual, mulai dari aktivitas awal, proses yang berlangsung, percabangan keputusan,

hingga aktivitas berakhir. Diagram ini membantu pengembang dan pengguna dalam memahami alur kerja sistem secara terstruktur, sehingga proses yang terjadi di dalam sistem dapat dianalisis dan dirancang dengan lebih mudah. Selain itu, *Activity Diagram* juga berfungsi untuk memodelkan proses bisnis dan menunjukkan hubungan antaraktivitas yang dilakukan oleh pengguna maupun sistem. Dengan menggunakan *Activity Diagram*, setiap tahapan proses dapat divisualisasikan secara jelas sehingga memudahkan identifikasi alur kerja, potensi kesalahan proses, serta kebutuhan sistem yang harus dipenuhi. Diagram ini juga membantu dalam proses dokumentasi dan pengembangan sistem karena memberikan gambaran yang rinci mengenai urutan aktivitas yang terjadi pada setiap fitur yang dirancang. Simbol-simbol yang digunakan dalam *Activity Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol activity diagram (Malius & Dani, 2021)

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain .
2		<i>Action</i>	<i>Action</i> menunjukkan suatu proses yang sedang dijalankan dalam sistem.
3		<i>Initial Node</i>	Bagaimana suatu proses sistem diawali.
4		<i>Activity Final Node</i>	Menunjukkan kondisi akhir dari suatu aktivitas, dimana seluruh proses telah selesai diproses.
5		<i>Fork Node</i>	Satu proses yang pada kondisi tertentu melakukan percabangan menjadi beberapa proses.

d. *Class Diagram*

Menurut Ramdany dkk. (2024), *Class Diagram* merupakan diagram yang digunakan untuk menggambarkan struktur sistem melalui hubungan antar kelas beserta atribut, metode, dan tanggung jawab yang dimiliki oleh setiap kelas. Diagram ini membantu dalam memahami interaksi antar entitas

serta menjadi gambaran rancangan basis data dan struktur objek yang digunakan dalam sistem yang akan dikembangkan. Selain itu, *Class Diagram* dapat menunjukkan hubungan antar kelas, sehingga memudahkan pengembang dalam memahami keterkaitan setiap komponen dalam sistem. Diagram ini juga berfungsi sebagai acuan dalam proses pembuatan basis data dan pengembangan program karena mampu menggambarkan data yang dikelola serta hubungan yang terjadi antar data dalam sistem. Dengan adanya *Class Diagram*, proses perancangan dan pengembangan sistem dapat dilakukan secara lebih terstruktur, meminimalkan kesalahan desain, serta memudahkan proses pemeliharaan dan pengembangan sistem di masa mendatang. Simbol-simbol dalam *Class Diagram* dapat disajikan pada Tabel 2.4.

Tabel 2.4 Simbol *class diagram* (Malius & Dani, 2021)

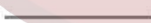
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Menunjukkan hubungan pewarisan antara kelas induk dan kelas turunan, sehingga kelas turunan mewarisi atribut dan metode dari kelas induk.
2		<i>Class</i>	Menunjukkan sekumpulan objek yang memiliki atribut, metode, dan karakteristik yang sama.
3		<i>Realization</i>	Menunjukkan hubungan antara antarmuka (<i>interface</i>) dan kelas yang mengimplementasikan antarmuka tersebut.
4		<i>Dependency</i>	Menunjukkan hubungan ketergantungan antara dua kelas, di mana perubahan pada satu kelas dapat memengaruhi kelas lainnya.

e. *Component Diagram*

Sibolon dkk., (2024) dalam penelitiannya yang berjudul *Research And Community Service Information System At LPPM Universitas MBP*

Medan With *Unified Modeling Language* (UML) menguraikan bahwa component diagram digunakan untuk menggambarkan komponen fisik dari suatu sistem perangkat lunak dengan memberikan penjelasan mengenai interaksi berbagai komponen yang terhubung satu sama lain. Component diagram juga membantu dalam memahami struktur arsitektur sistem sehingga memudahkan dalam proses perancangan, pengembangan, serta pemeliharaan sistem. Simbol-simbol dalam component diagram dapat disajikan pada Tabel 2.5.

Tabel 2.5 Simbol *component diagram* (Sagala, 2024)

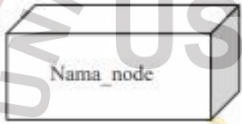
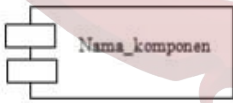
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Package</i>	Menunjukkan kumpulan komponen atau elemen yang dikelompokkan dalam satu paket berdasarkan fungsinya.
2		<i>Component system</i>	Menunjukkan komponen sistem yang berisi fungsi atau layanan tertentu dalam aplikasi.
3		<i>Dependency</i>	Menunjukkan hubungan ketergantungan antar komponen, di mana perubahan pada satu komponen dapat memengaruhi komponen lainnya.
4		<i>Link</i>	Menunjukkan hubungan atau koneksi antar komponen dalam sistem.

f. *Deployment Diagram*

Penelitian yang dilakukan oleh Akbar dan Azis (2024) dengan judul Implementasi Model View Controller Untuk Proses Pencatatan dan Monitoring Pelanggaran Siswa Pada SMK Satria menguraikan bahwa *deployment diagram* menggambarkan bagaimana perangkat keras dan perangkat lunak saling berinteraksi dengan memvisualisasikan berbagai komponen sistem, seperti server, database, dan klien. *Deployment diagram*

membantu pengembang dalam memahami lokasi penyimpanan data secara terperinci. Diagram ini juga menggambarkan interaksi antar pengguna dengan aplikasi berbasis web yang akan dibangun. Interaksi antar pengguna dengan aplikasi web dapat berupa bagaimana permintaan pengguna dikirim, permintaan pengguna diproses, dan permintaan pengguna dikembalikan dengan bentuk respons yang sesuai. Simbol-simbol dalam *deployment* diagram dapat disajikan pada Tabel 2.6.

Tabel 2.6 Simbol *deployment* diagram (Rohmah & Voutama, 2024)

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Package</i>	Menunjukkan kumpulan node atau elemen yang dikelompokkan dalam satu paket.
2		<i>Node</i>	Menggambarkan komponen apa saja yang terdapat pada sistem.
3		<i>Dependency</i>	Menunjukkan hubungan ketergantungan antar node, di mana perubahan pada satu node dapat memengaruhi node lainnya.
4		<i>Communicates/Link</i>	Menghubungkan antar node yang saling berinteraksi.
5		<i>Component</i>	Menggambarkan elemen yang terdapat pada suatu node.

2.3.10 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi dan kebutuhan serta menemukan kesalahan yang terjadi pada sistem perangkat lunak untuk diperbaiki guna menjamin kualitas dari perangkat lunak (Uminingsih dkk., 2022). Pengujian sistem dilakukan dengan tujuan untuk memastikan bahwa sistem berfungsi

dengan baik dan layak untuk diimplementasikan secara penuh oleh pengguna.

Pengujian sistem akan dilakukan melalui dua tahapan yaitu teknik *black box testing* dan *Beta Testing*. *Black box testing* merupakan teknik pengujian sistem yang difokuskan pada fungsionalitas sistem (Febriyantia dkk., 2021). *Black box testing* memiliki keunggulan dalam hal mengidentifikasi aspek-aspek yang tidak sesuai dengan spesifikasi kebutuhan yang telah ditentukan. Selain memiliki keunggulan, *black box testing* juga memiliki kelemahan yang terletak pada keterbatasan pengujian yang tidak dapat dilakukan secara menyeluruh (Praniffa dkk., 2023).

Menurut Samdono dkk., (2024) menguraikan bahwa kalkulasi persentase kelayakan fungsional sistem diperoleh dari rasio perbandingan antara jumlah skenario data uji yang berstatus valid terhadap total keseluruhan skenario pengujian yang telah direncanakan sebelumnya. Rumus matematis untuk menghitung tingkat keberhasilan fungsional tersebut diformulasikan pada persamaan (2.1) dan persamaan (2.2).

$$\text{Nilai Efektivitas Tabel} = \left(\frac{\sum \text{Pengujian Sesuai}}{\sum \text{Jumlah Skenario Pengujian}} \right) \times 100 \% \quad (2.1)$$

$$\text{Nilai Efektivitas Website} = \left(\frac{\text{Nilai Tabel A} + \text{Nilai Tabel B} + \dots + \text{Nilai Tabel n}}{\sum \text{Jumlah Tabel Yang Diuji}} \right) \quad (2.2)$$

Melalui E-book yang berjudul Pengujian Perangkat Lunak Selenium IDE dan GTMetrix karya Yohanes Jaunardi Palis dan Ryan Putranda Kristianto yang melakukan penelitian pada website Healthy Choice Indonesia menguraikan bahwa pengujian *black box testing* pada website dilakukan dengan cara memasukkan input yang tidak valid dan mencoba berbagai situasi untuk melihat bagaimana website tersebut merespons. Daftar fitur-fitur atau halaman yang diuji dari website Healthy Choice meliputi halaman register, halaman login, fitur lupa password, fungsionalitas menu pada halaman home page, fitur pencarian, fitur log out, fitur chat, dan fitur keranjang. Berdasarkan hasil pengujian beberapa fitur di atas, Penulis tidak menemukan pengujian yang tidak sesuai dengan tidak diharapkan atau tidak valid (Palis & Kristianto 2024).

Beta Testing merupakan metode pengujian yang melibatkan sejumlah kecil pengguna yaitu 44 responden untuk melakukan uji coba pada aplikasi. Pengujian ini dilakukan agar kualitas produk dapat meningkat dengan memperbaiki bug atau kesalahan pada sistem (Irawan dkk., 2024). *Beta Testing* mampu memberikan pemahaman mengenai masalah tak terduga yang mungkin akan timbul selama penggunaan sistem. Informasi ini sangat berguna dalam menentukan fungsi dasar dari sistem tersebut. Pada pengujian *Beta Testing* ini, peneliti perlu menentukan pengguna yang akan terlibat dalam pengujian serta menetapkan skenario pengujiannya (Yakub dkk., 2024).

Pada E-book yang berjudul *Perancangan dan Pengembangan Sistem Informasi Berbasis Scrum* yang ditulis oleh Mohd. Amiruddin Saddam menguraikan bahwa pengujian *Beta Testing* merupakan pengujian dengan cara penerapan langsung perangkat lunak di lingkungan yang tidak dapat sepenuhnya dikendalikan oleh pengembang. Perangkat lunak memasuki tahap beta ketika sebagian besar fungsinya telah diaktifkan. Pada tahap ini, perangkat lunak diuji untuk memungkinkan pengguna mengoperasikan perangkat lunak, mengidentifikasi bug, dan memberikan masukan agar masalah tersebut dapat diperbaiki sebelum produk dirilis ke pengguna akhir (Saddam, 2024). Pengujian *Beta Testing* dimulai dengan beberapa tahapan, yaitu menyusun kuesioner dan membagikannya ke responden, responden memberikan jawaban, data yang diperoleh kemudian dianalisis menggunakan angket dengan skala likert. Menurut (Hakim dkk., 2024). Jawaban tersebut dihitung dengan rumus pada persamaan (2.3) dan (2.4)

$$X = \sum(N \times R) \quad (2.3)$$

$$Y = \left(\frac{X}{\text{Skor Ideal}} \right) \times 100 \% \quad (2.4)$$

$$\text{Skor Ideal} = \text{nilai likert tertinggi} \times \text{jumlah responden}$$

Keterangan:

Y = nilai persentase yang dicari

X = jumlah dari hasil perkalian setiap jawaban responden

N = nilai dari jawaban responden

R = jumlah responden

Tabel 2.7 Skala Penilaian (Hakim dkk., 2024)

SKALA PENILAIAN		
No	Keterangan	Skala (N)
1.	Sangat Tidak Setuju	1
2.	Tidak Setuju	2
3.	Netral	3
4.	Setuju	4
5.	Sangat Setuju	5

Tabel 2.8 Pertanyaan *Beta Testing* (Hakim dkk., 2024)

NO	PERTANYAAN
1.	Apakah sistem informasi pelayanan masyarakat Kelurahan Jantiharjo dapat berjalan dengan baik?
2.	Apakah tampilan website mudah dipahami dan digunakan oleh pengguna?
3.	Apakah proses login pada sistem berjalan dengan baik?
4.	Apakah menu pengajuan surat mudah diakses dan berjalan dengan baik?
5.	Apakah menu pengaduan mudah diakses dan berjalan dengan baik?
6.	Apakah fitur pengecekan status permohonan berjalan dengan baik?
7.	Apakah informasi profil, visi misi, dan struktur organisasi kelurahan tersaji dengan jelas?
8.	Apakah sistem dapat membantu meningkatkan kepuasan pelayanan di Kelurahan Jantiharjo?
9.	Apakah sistem mampu mempercepat proses pelayanan di Kelurahan Jantiharjo dibandingkan cara manual?
10.	Apakah anda merasa puas terhadap penggunaan sistem informasi pelayanan masyarakat ini?

Menurut Akbar & Nasrulloh (2025) menguraikan bahwa persentase penilaian diklasifikasikan menjadi lima kategori, mulai dari Sangat Tidak Setuju hingga Sangat Setuju. Kriteria interpretasi tersebut dapat disajikan pada Tabel 2.9.

Tabel 2.9 Kriteria Persentase *Beta Testing* (Akbar & Nasrulloh, 2025)

No	Persentase	Kriteria Kesimpulan (N)
1.	0% – 20%	Sangat Tidak Setuju
2.	21% – 40%	Tidak Setuju
3.	41% – 60%	Netral
4.	61% – 80%	Setuju
5.	81% – 100%	Sangat Setuju

