

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Sebelumnya, telah ada beberapa penelitian terdahulu yang membahas tentang pembuatan surat dengan berbasis website, penelitian yang pertama menjelaskan bahwa pengembangan aplikasi dalam pengelolaan surat masuk dan keluar berbasis web, banyak ditemukan bahwa pengelolaan surat secara manual menimbulkan berbagai kendala, termasuk keterlambatan tindak lanjut surat dan penyampaian informasi. Salah satu penelitian pada Dinas Pertanian Pangan dan Perikanan Kabupaten Karanganyar menunjukkan bahwa aplikasi pengelolaan surat berbasis web ini dapat mempermudah pencarian data dan pengelolaan surat masuk maupun keluar, yang secara langsung meningkatkan efisiensi operasional instansi. Metode pengembangan yang digunakan, yaitu Waterfall, dimana dalam perancangan sistem yang terstruktur dari analisis hingga implementasi, dilengkapi dengan pemodelan Unified Modeling Language (UML) untuk memetakan proses bisnis secara visual. Aplikasi ini juga dilengkapi berbagai fitur utama seperti pencarian data, disposisi surat masuk, pengajuan surat keluar, dan pelaporan otomatis, yang kesemuanya dirancang untuk mempercepat proses administratif dan meningkatkan akurasi serta aksesibilitas data surat. Berdasarkan uji kepuasan pengguna menggunakan User Acceptance Test (UAT), penelitian ini melaporkan tingkat kepuasan sebesar 79%, menunjukkan bahwa aplikasi ini diterima dengan baik dan memenuhi kebutuhan pengguna dalam mengelola data surat dengan lebih sistematis dan efisien. Temuan ini konsisten dengan penelitian sejenis yang mengindikasikan bahwa aplikasi berbasis web membantu mempercepat proses pencarian dan pemrosesan data, mengurangi risiko keterlambatan, dan meningkatkan transparansi dalam pengelolaan surat di berbagai instansi pemerintahan Widya Nur'Afnida, dkk (2024:1030-1040).

Menurut Yunitasari, dkk (2023:81-91) menjelaskan bahwa penelitian pada sistem komputerisasi administrasi surat di program studi teknik Informatika

universitas sahid surkart. Dengan mengembangkan Sistem Komputerisasi Administrasi Surat menggunakan bahasa pemrograman PHP dan basis data MySQL yang mengintegrasikan fitur-fitur penting seperti upload dan *download* surat masuk, serta manajemen data master mahasiswa dan Kaprodi. Sistem ini dibangun untuk mengatasi permasalahan yang ada, seperti error pada halaman login dosen dan tampilan menu yang kurang baik, sehingga mampu meningkatkan pengalaman pengguna. Hasil uji coba menggunakan metode *black box* menunjukkan bahwa sistem ini berfungsi dengan baik dan tidak terdapat kendala berarti selama penggunaan normal, yang menghasilkan tingkat kepuasan tinggi di kalangan pengguna. Temuan ini menekankan bahwa sistem informasi yang dirancang sesuai dengan kebutuhan spesifik pengguna, serta dilakukan perbaikan berkelanjutan, akan memberikan dampak positif terhadap efektivitas proses surat-menyurat dalam konteks akademik.

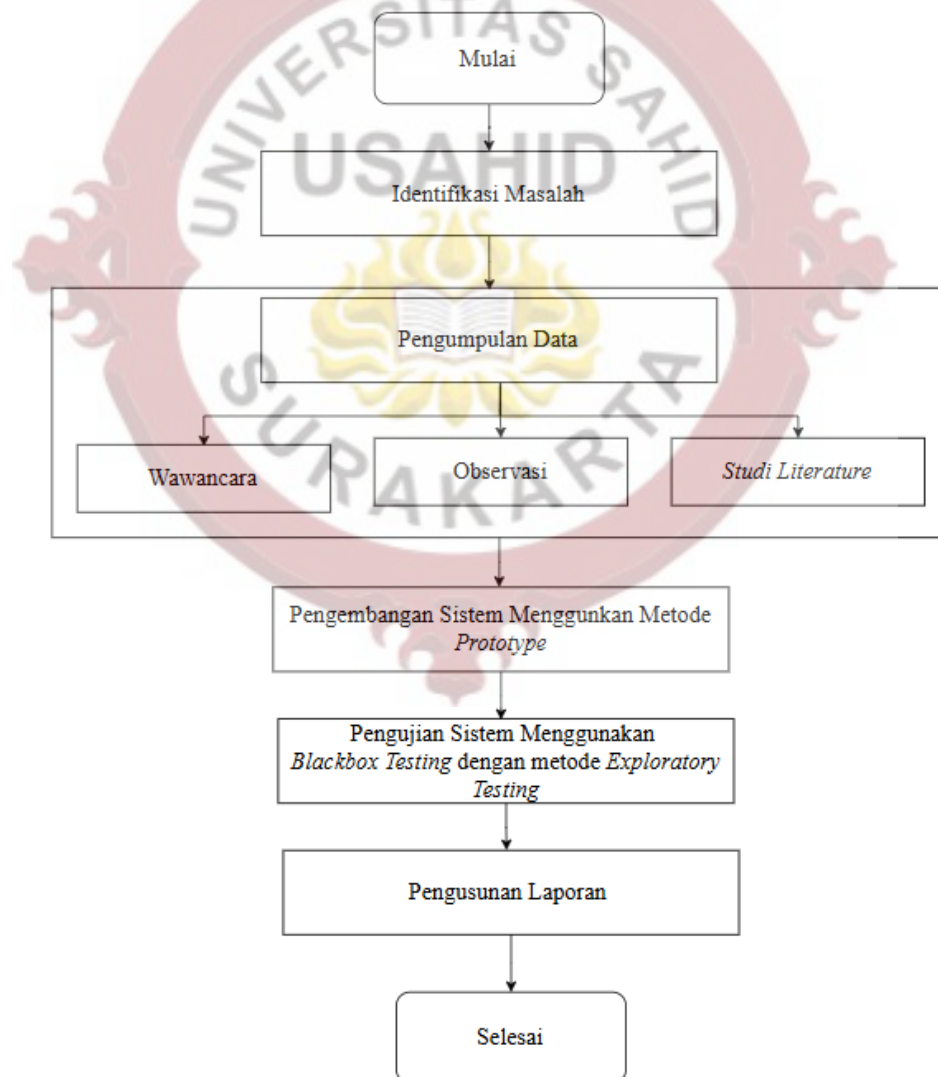
Menurut Raden Erwin G.R dan Parid Ma'rup (2021:25- 24) menyatakan bahwa proses pelayanan kepada masyarakat di desa dalam pelayanan publik masih dilakukan secara konvensional, yaitu masyarakat datang ke kantor desa dengan membawa surat pengantar dari RT yang diketahui oleh RW serta membawa berkas yang dibutuhkan untuk pembuatan surat-surat. Kemudian, masyarakat menunggu giliran untuk dilayani oleh operator desa. sistem informasi pelayanan administrasi publik desa berbasis web yang mampu menjembatani antara penduduk dengan perangkat desa dalam melakukan pengajuan surat kependudukan secara online, sehingga mempersingkat proses birokrasi. Sistem yang dibangun dapat mengorganisir arsip dengan lebih tertata dibandingkan sebelumnya.

Menurut Miswar Papuangan, dkk (2023:93-106) menjelaskan bahwa sistem pelayanan administrasi desa berbasis web di desa menggunakan Model *SDLC* untuk proses pembuatan sistem dan metodologi untuk merekayasa perangkat lunak. Serta alat bantu lain yang digunakan untuk merancang sistem yang dibangun berupa diagram alir, diagram konteks, data *flow diagram*, serta menggunakan bahasa pemrograman *PHP*, database menggunakan *MySQL*. Pengujian sistem administrasi Desa Daruba menggunakan teknik *black box testing*.

Black box testing Hasil yang diperoleh dari rancangan dan pengujian yang telah dilakukan, sistem administrasi Desa Daruba dapat membantu masyarakat Desa Daruba dalam melakukan pengajuan pembuatan surat keterangan melalui website yang telah dibangun serta membantu memudahkan kepala desa dan staf desa dalam pengeolaan administrasi pembuatan surat yang lebih baik, efektif, dan efisien.

2.2 Kerangka Pemikiran

Kerangka pemikiran pada judul “Rancang Bangun Aplikasi Pelayanan Masyarakat Untuk Pembuatan Surat Keterangan Secara Online di Kantor Desa Saradan”. Disajikan pada Gambar 2.1.



Gambar 2.1 Kerangka Pemikiran

Setiap kerangka pemikiran dijelaskan pada Gambar 2.1 sebagai berikut ini:

1. Identifikasi Masalah

Melakukan identifikasi masalah tentang masalah apa yang akan dibahas yang berkaitan dengan sistem pelayanan masyarakat dalam pembuatan surat keterangan yang ada pada Kantor Desa Saradan Kecamatan Karangmalang, Kabupaten Sragen, Provinsi Jawa Tengah.

2. Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

a. Interview atau Wawancara.

Teknik pengumpulan data dengan mengadakan tanya jawab secara langsung dengan staf pelayanan pembuatan surat keterangan pada Desa Saradan Kecamatan Karangmalang, Kabupaten Sragen. Untuk memperoleh data-data yang dijadikan sebagai bahan penelitian.

b. Observasi

Teknik pengumpulan data dengan mengadakan penelitian langsung pada Desa Saradan Kecamatan Karangmalang, Kabupaten Sragen. untuk meninjau langsung permasalahan yang terjadi pada Kantor Desa Saradan tersebut.

c. Studi Literatur

Pengumpulan data dengan cara *literature*, jurnal, paper dan bacaan-bacaan yang berkaitan dengan sistem pendukung keputusan.

3. Pengembangan Sistem

Pada penelitian ini pengembangan sistem berbasis website dengan menggunakan metode *Prototype*. berikut adalah tahapan pengembangan sistem menggunakan metode *prototype*:

a. Communication

Bertemu langsung dengan staf pelayanan masyarakat pada Kantor Desa Saradan dan menentukan tujuan umum serta kebutuhan yang diinginkan dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.

b. Quick Plan

Tahapan ini dilakukan dengan perancangan secara cepat, dengan menambahkan beberapa fungsi dan fitur dari sebuah sistem dalam proses pengajuan surat keterangan secara *online* dengan spesifikasi untuk pengembangan berdasarkan kebutuhan sistem, sesuai hasil komunikasi yang dilakukan agar pengembangan dapat sesuai yang diharapkan.

c. *Modelling Quick Design*

Tahap *modeling quick design*, fokus yang dilakukan adalah merancang aspek-aspek yang terlihat oleh pengguna akhir. Dengan merancang antarmuka pengguna *user interface* dalam bentuk suatu format tampilan. Model aplikasi digambarkan dalam bentuk *UML* yang meliputi *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, *Sequence diagram* dan *User Interface*.

d. *Construction of Prototype*

Tahapan ini digunakan untuk membangun *prototype* dan menguji-coba sistem yang dibangun. Proses *instalasi* dan penyediaan *user support* juga dilakukan agar sistem dapat berjalan dengan sesuai.

e. *Deployment Delivery & Feedback*

Tahapan ini digunakan untuk mendapatkan *feedback* dari staf pelayanan Kantor Desa Saradan terhadap aplikasi pembuatan surat keterangan secara online, sebagai hasil *evaluasi* dari tahapan sebelumnya dan *implementasi* dari sistem yang dikembangkan.

4. Pengujian Sistem

Pengujian sistem dilakukan menggunakan *blackbox* testing dengan metode *Exploratory Testing*. menguji kualitas sistem yang dibuat serta memastikan fungsionalitas, khususnya pada input dan output aplikasi apakah sudah sesuai dengan apa yang diharapkan atau belum. Dan pengujian pemberian kuesioner kepada pengguna yang berisi sepuluh pernyataan tentang pengalaman mereka menggunakan aplikasi tersebut.

5. Penyusunan Laporan

Pada tahap ini dilakukan penyusunan laporan dari hasil penelitian dari awal

sampai akhir tentang sistem pembuatan surat keterangan secara *online* yang telah dilakukan pada Kantor Desa Saradan.

2.3 Teori Pendukung

Teori pendukung digunakan sebagai landasan teori untuk memperkuat dan memvalidasi argumen utama yang diajukan dalam penelitian yang diambil dari *literatur* atau penelitian terdahulu yang sudah diakui keabsahannya, bertujuan untuk memberikan landasan ilmiah yang kokoh pada penelitian.

2.3.1 Kualitas Pelayanan

Menurut Suwithi dan Anwar (2020:81-88) kualitas pelayanan adalah mutu dari pelayanan yang diberikan kepada masyarakat, baik masyarakat internal maupun masyarakat eksternal berdasarkan standar prosedur pelayanan. Kemudian menurut Kotler (2022:1023-1028), kualitas pelayanan merupakan totalitas dari bentuk karakteristik barang dan jasa yang menunjukkan kemampuannya untuk memuaskan kebutuhan pelanggan, baik yang nampak jelas maupun tersembunyi. Bagi perusahaan yang bergerak di sektor jasa, pemberian pelayanan yang berkualitas pada masyarakat merupakan hal mutlak yang harus dilakukan apabila perusahaan ingin mencapai keberhasilannya. Dan menurut Sulastiyono (2019:7-9), menjelaskan bahwa kualitas pelayanan adalah sesuatu yang kompleks, dan tamu akan menilai kualitas pelayanan melalui lima prinsip dimensi pelayanan sebagai ukurannya, yaitu Reabilitas, Responsif, Kepastian atau Jaminan, Empati, dan Nyata.

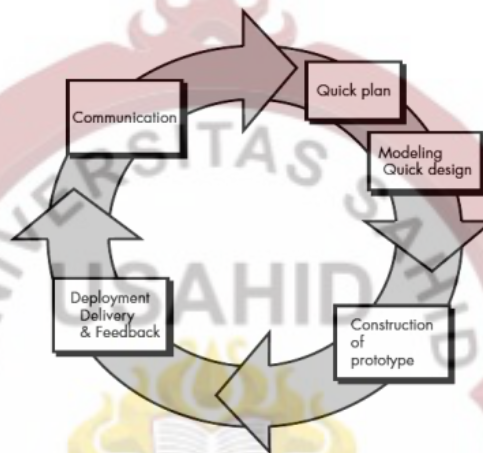
2.3.2 Sistem Pelayanan Administrasi

Menurut Rahmawati dan Fatmawati (2020:134-140) sistem pelayanan administrasi merupakan suatu kegiatan pengelolaan, pengumpulan dan pelaporan data yang berupa pembukuan, ketik mengetik, pemaparan agenda dan surat menyurat. Pelayanan surat menyurat merupakan salah satu kegiatan yang penting disetiap desa. Terdapat beberapa jenis surat yang biasanya dilayani oleh desa seperti surat keterangan pindah, surat domisili, surat keterangan serbaguna, surat keterangan kematian, surat keterangan tidak mampu, surat keterangan berkelakuan baik, surat keterangan usaha, surat keterangan belum menikah dan

beberapa surat-surat lainnya.

2.3.3 Metode Pengembangan *Prototype*

Menurut Wicaksana, dkk (2019:83-86) metode *prototype* merupakan suatu paradigma baru dalam metode pengembangan perangkat lunak dimana metode ini tidak hanya sekedar *evolusi* dalam dunia pengembangan perangkat lunak, tetapi juga *merevolusi* metode pengembangan perangkat lunak yang lama yaitu sistem *sekuensial* yang biasa dikenal dengan nama SDLC. Disajikan pada Gambar 2.2.



Gambar 2.2 Contoh Model Prototype

2.3.4 *Unified Modeling Language* (UML)

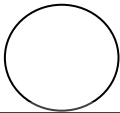
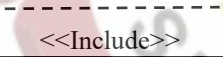
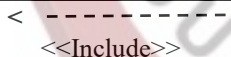
Menurut Windu Gata (2021:115-121), *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* dapat disajikan pada Tabel 2.1

Sebagai berikut ini.

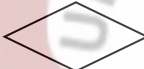
Tabel 2.1 Simbol *Use Case Diagram*

Gambar Simbol	Keterangan
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
	Aktor atau Aktor adalah Abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i> , tetapi tidak memiliki kontrol terhadap use case
	<i>Asosiasi</i> antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	Include, merupakan di dalam use case lain (<i>required</i>) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari use case lain jika kondisi atau syarat.

2. *Activity Diagram*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat disajikan pada Tabel 2.2

Tabel 2.2 Simbol *Activity Diagram*

Gambar Simbol	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>End Point</i> , akhir dari aktivitas.
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Join (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, true atau false
	Swimlane, pembagian activity diagram untuk menunjukkan siapa melakukan apa.

3. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram* dapat disajikan pada Tabel 2.3

Tabel 2.3 Simbol *Sequence Diagram*

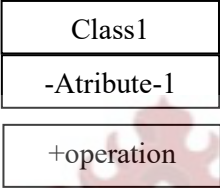
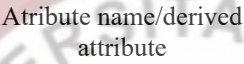
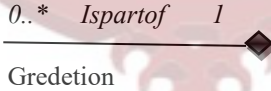
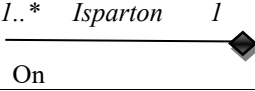
Gambar Simbol	Keterangan
	Entity Class, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	Boundary Class, berisi kumpulan kelas yang menjadi interfaces atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak.
	Control class, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	Message, simbol mengirim pesan antar class.
	Recursive, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	Activation, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.

4. *Class Diagram*

Class Diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan. Simbol-simbol yang digunakan dalam *Class Diagram* dapat disajikan pada Tabel 2.4 Sebagai

berikut ini.

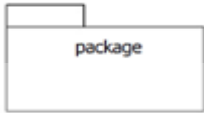
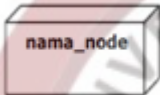
Tabel 2.4 Simbol *Class Diagram*

Nama	Simbol	Deskripsi
<i>Class</i>		Berisi jenis orang, tempat atau hal yang perlu disimpan oleh sistem. Memiliki nama yang diketik tebal di bagian tengah atas, dilanjutkan dengan daftar atribut pada kompartemen tengah dan daftar operasi pada kompartemen bawah.
<i>Attribute</i>		Merepresentasikan properti yang mendeskripsikan status objek. Dapat diturunkan dari atribut lain yang ditunjukkan dengan menempatkan garis miring sebelum nama atribut.
<i>Association</i>		Hubungan antara banyak kelas dan dirinya sendiri. Label adalah kata kerja atau nama peran yang mewakili hubungan. Bisa ada di antara satu kelas atau lebih. Berisi simbol yang mewakili waktu minimum dan maksimum kelas yang saling berhubungan dengan kelas lainnya.
<i>Generalization</i>		Hubungan antara satu kelas dengan kelas lainnya.
<i>Aggregation</i>		Merupakan bentuk khusus dari <i>association</i> yang menggambarkan bagian dari hubungan <i>logical</i> antara sebuah kelas atau lebih dengan dirinya sendiri.
<i>Composition</i>		Merupakan bentuk khusus dari <i>association</i> yang menggambarkan bagian dari hubungan fisik antara sebuah kelas atau lebih dengan dirinya sendiri.

5. *Deployment Diagram*

Deployment Diagram menunjukkan arsitektur fisik pada hardware dan software pada suatu sistem yang dirancang. *Deployment diagram* juga dapat menunjukkan perangkat - perangkat dan nodes diantara hubungan yang dimilikinya antar komponen.

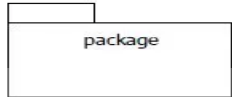
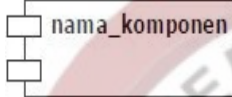
Tabel 2.5 Simbol *Deployment Diagram*

Gambar Simbol	Keterangan
	Package adalah sebuah bungkusan dari satu atau lebih node.
	Node biasanya mengacu pada hardware, perangkat lunak yang tidak dibuat sendiri (software). Jika didalam node disertai komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.
	Dependency adalah kebergantungan antar node, arah panah mengarah pada node yang dipakai.
	Link merupakan relasi antar node.

6. *Component Diagram*

Component Diagram menunjukkan struktur dan hubungan antar komponen software termasuk ketergantungan (dependency) diantara komponen-komponen tersebut. Komponen pada piranti lunak adalah berupa modul-modul yang berisikan kode, baik library maupun executable. Umumnya komponen yang terbentuk dari beberapa class dan/atau package, atau juga dapat dari komponen-komponen yang lebih kecil

Tabel 2.6 Simbol *Component Diagram*

Gambar Simbol	Keterangan
	Package merupakan simbol yang berperan untuk mewadahi atau sebagai wadah dari komponen
	Componen system: merupakan simbol yang digunakan untuk memberikan gambaran terkait perangkat keras atau hardware maupun objek yang ada di dalam sistem.
	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipaka.
	Sama dengan konsep interface pada pemrograman berorientasi objek, yaitu sebagai antar muka komponen anggar tidak mengakses komponen langsung.
	Relasi antar komponen.

2.3.5 Balsamiq Mockup 3

Menurut (Munawar dkk, 2019) *Balsamiq mockup 3* adalah *software* yang digunakan untuk pembuatan tampilan antarmuka pengguna atau *user interface* sebuah aplikasi. *Software* ini merupakan salah satu aplikasi yang banyak digunakan oleh para perancang aplikasi yang dimana perangkat lunak ini sudah menyediakan alat yang dapat memudahkan dalam membuat desain *prototyping* aplikasi yang akan kita buat. *Software* ini berfokus pada konten yang ingin di

gambar dan fungsionalitas yang dibutuhkan oleh pengguna. Kelebihan *Balsamiq Mockups 3* dibanding *software* pembuat *mockup* lainnya adalah aplikasi ini berbasis *cloud*, disertai aplikasi desktop yang memungkinkan kita dengan cepat dan mudah membuat rancangan *website*.

2.3.6 Website

Menurut Maulana dkk., (2021:17-21) *Website* adalah sebuah perkumpulan laman yang bertujuan untuk dapat menampilkan sebuah informasi dimana informasi tersebut dalam bentuk tulisan, gambaran dan suara yang mempunyai sifat baik tetap ataupun berubah dan akan membentuk sebuah rangkaian yang terhubung dengan masing-masing jaringan laman yang dihubungkan. Sedangkan menurut Abdulloh dalam menyatakan bahwa *website* merupakan sebuah perkumpulan halaman yang terdapat dari beberapa laman yang mempunyai suatu informasi yang berbentuk digital dengan cara disajikan melewati jalur koneksi sebuah internet dan *website* ini biasanya disingkat dengan *web*.

2.3.7 Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*. Teks editor ini secara langsung mendukung bahasa pemrograman *JavaScript*, *Typescript*, dan *Node.js*, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang via marketplace *Visual Studio Code* (seperti *C++*, *PHP*, *C#*, *Python*, *Go*, dan *Java* Permana dan Romadlon (2019: 213-219).

2.3.8 Hypertext Markup Language (HTML)

Menurut Samah Hartanto (2021:21-26) *HTML* singkatan dari *Hyper Text Markup Language*, yaitu *skrip* yang berupa *tag-tag* untuk membuat dan mengatur struktur *website*. Oleh karena itu pengetahuan tentang *HTML* sangat penting dalam membangun aplikasi web” Abdurahman Hidayat, dkk (2022:41-51). Mira Orisa, mengemukakan bahwa “*HTML* merupakan dasar dari semua konten yang muncul di *World Wide Web (WWW)*, terdiri dari dua bagian penting yaitu konten

informasi dan satu set instruksi yang memberi tahu komputer Anda cara menampilkan konten tersebut”.

2.3.9 *Hipertext Proprocessor (PHP)*

PHP (*Hipertext Proprocessor*) merupakan bahasa pemrograman yang bersifat *server side*, PHP dirancang khusus untuk mengisi kesenjangan antara halaman HTML *statis* dan halaman penuh dinamis, *skrip* PHP akan dituliskan dalam *editor teks* dasar” Rusli, dkk (2019). Bahasa pemrograman PHP yang umum dipakai dalam pembuatan dan pengembangan suatu web, yang dilansir dari PHP.net, PHP merupakan singkatan dari *PHP: Hypertext Preprocessor*, menurut situs tersebut, PHP adalah bahasa pemrograman yang banyak digunakan secara luas dan secara khusus sesuai untuk pengembangan web yang sebelum dinamakan demikian, PHP artinya adalah *Personal Home Page Tools* karena dipakai membangun web pribadi. jadi PHP bukan lagi sekadar dipakai membangun halaman web sederhana, tetapi membuat berbagai web ternama dunia seperti *WordPress* dan *Wikipedia* Tandika, dkk (2023:1-11).

2.3.10 *Cascading Style Sheet (CSS)*

Menurut Andy Antonius S, dkk (2019:1-9) *Cascading style sheets (CSS)* adalah suatu bahasa pemrograman yang digunakan untuk mendukung pembuatan website agar memiliki tampilan yang lebih menarik dan terstruktur. Berfungsi mengatur elemen HTML dengan berbagai properti yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan. Tujuannya tak lain untuk mempermudah proses penataan halaman web Sama seperti HTML, CSS juga berkembang dari waktu ke waktu seiring dengan pesatnya perkembangan zaman maka CSS itu banyak versinya Berikut ini adalah beberapa versi dari perkembangan CSS dari awal sampai sekarang:

2.3.11 *Xampp*

Menurut Hidayat, dkk (2019:12-15) XAMPP adalah sebuah aplikasi web *server* instan dan lengkap dikarenakan segala yang butuhkan untuk membuat sebuah situs web dengan *Content Management System*, dan XAMPP merupakan

sebuah paket *installer* AMP (Apache, MySQL, dan Php) yang sangat mudah untuk diaplikasikan dalam komputer yang belum memiliki *server* untuk dapat melihat situs yang dibuat menggunakan bahasa *server*, dan *database* server tersebut.

2.3.12 JavaScript

Menurut Sholikhah mahdi S.H (2022:21-28) *JavaScript* adalah bahasa pemrograman *web* yang digunakan untuk meningkatkan pengalaman pengguna saat menggunakan halaman *web*. Meskipun memiliki nama yang mirip, *JavaScript* tidak ada hubungannya dengan bahasa pemrograman Java. *JavaScript* disertakan dengan hampir semua *browser web* dan tidak memerlukan perangkat lunak tambahan untuk dijalankan. Penggunaan *JavaScript* dapat membuat halaman *web* menjadi lebih interaktif dan responsif, seperti mengisi formulir secara otomatis atau menambahkan berbagai fitur yang meningkatkan kegunaan. Program *JavaScript* berjalan di *browser web* pengguna, yang memiliki kelebihan dan kekurangan. Meskipun secara teori semua *JavaScript* harus berjalan sama, dalam praktiknya, *browser* yang berbeda dapat menafsirkan *JavaScript* dengan cara yang berbeda, terutama pada versi *browser* yang lebih lama.

2.3.13 Pengujian Blackbox Testing

Menurut Nuris (2022:1-5) Pengujian *blackbox* (*blackbox testing*) adalah salah satu metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas, khususnya pada *input* dan *output* aplikasi (apakah sudah sesuai dengan apa yang diharapkan atau belum). Tahap pengujian merupakan salah satu tahap yang harus ada dalam sebuah siklus pengembangan perangkat lunak. ciri-ciri *black box testing*, diantaranya sebagai berikut:

1. *Black box testing* berfokus pada kebutuhan fungsionalitas pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
2. *Black box testing* bukan teknik alternatif dari pada *white box testing*. Lebih dari pada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*.

3. *Black box testing* melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. Juga disebut sebagai *behavioral*.

2.3.14 Pengujian Kuesioner

Pengujian sistem aplikasi menggunakan kuesioner bisa dilakukan untuk mengukur kepuasan pengguna. Kuesioner merupakan alat riset yang berisi serangkaian pertanyaan tertulis untuk mendapatkan tanggapan dari responden (Slamet dan Wahyuningsih, (2022: 51-58).

Berikut beberapa hal yang perlu diperhatikan saat menggunakan kuesioner untuk pengujian sistem aplikasi:

1. Validitas kuesioner dikatakan valid jika dapat mengukur apa yang hendak diukur.
2. Desain kuesioner yang baik dapat memastikan data yang dikumpulkan akurat sehingga hasilnya dapat ditafsirkan dan digeneralisasikan.
3. Keuntungan Kuesioner memiliki beberapa keuntungan, seperti tidak memerlukan kehadiran peneliti, dapat diberikan secara serentak kepada banyak responden, dan dapat dibuat anonim.

Pengujian sistem aplikasi dilakukan untuk mengetahui seberapa baik dan sesuai sistem yang dibuat dengan kebutuhan yang telah ditetapkan. Pengujian perangkat lunak yang baik dapat mencegah bug dan meningkatkan kinerja.