

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Teori Umum

Di dalam sub bab berikut, penulis akan membahas mengenai teori-teori umum yang berhubungan dengan pelaksanaan penelitian ini yang berupa pengertian-pengertian secara umum.

2.1.1 Sistem

Sistem berasal dari bahasa Latin (*Systema*) dan Yunani (*Sustema*), yaitu suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memperlancar aliran informasi, material, ataupun energi. Terminologi ini sering digunakan untuk menggambarkan sekumpulan entitas yang berinteraksi (Febryan et al., 2020).

Sistem merupakan suatu gabungan dari beberapa komponen, entitas atau subsistem yang saling terhubung satu sama lain untuk mencapai suatu tujuan tertentu (Mulia, 2020). Menurut (Silalahi & Saragih, 2022), sistem dapat didefinisikan berdasarkan prosedurnya, yang menekankan bahwa sistem merupakan suatu jaringan dari prosedur yang berhubungan dan berkumpul satu sama lainnya dan melakukan serangkaian aktivitas untuk mencapai tujuan tertentu, dan sistem juga dapat didefinisikan berdasarkan elemennya, yang menekankan bahwa sistem merupakan kumpulan dari berbagai elemen yang saling berinteraksi untuk mencapai suatu tujuan. Sedangkan menurut (Ilmi & Metandi, 2020), sistem

merupakan suatu susunan yang sistematis dari berbagai kegiatan dan prosedur yang saling tergantung dan berhubungan untuk melaksanakan dan mempermudah kegiatan dalam organisasi.

Dari uraian-uraian yang ada diatas, maka penulis menyimpulkan bahwa sistem merupakan gabungan dari berbagai elemen, entitas, komponen maupun subsistem, dapat berupa nyata ataupun abstrak, yang terkumpul, terhubung dan saling tergantung satu sama lainnya dalam melakukan aktivitas-aktivitas tertentu untuk mencapai suatu tujuan tertentu.

2.1.2 Informasi

Informasi adalah data-data yang dikumpulkan menjadi satu, yang akan dipakai dalam pengambilan keputusan oleh penerima, dimana data tersebut telah dianalisis terlebih dahulu (Lesmana & Silalahi, 2022). Hal ini sejalan dengan pendapat (Ilmi & Metandi, 2020), yang menyatakan bahwa sumber dari suatu informasi ialah data, dimana data tersebut telah diproses menjadi bentuk yang memiliki fungsi bagi penerima informasi.

Pengolahan data menjadi informasi yang berguna bagi penerimanya, dapat menjadi suatu dasar untuk pengambilan keputusan, dimana konsekuensinya dapat dirasakan secara langsung atau tidak langsung dimasa yang akan datang (Elisa et al., 2020). Kualitas informasi itu sendiri tergantung dari tiga hal, yaitu informasi harus akurat, informasi harus tepat pada waktunya dan informasi harus relevan (Tukino & Amrizal, 2020).

2.1.3 Sistem Informasi

Sistem informasi merupakan suatu sistem yang terdapat di suatu lingkup atau area tertentu, untuk mendukung kegiatan-kegiatan yang dilakukan di lingkungan tersebut dan menghasilkan suatu laporan yang diperlukan atas kegiatan yang telah dilakukan (S. A. Kurniawan et al., 2022). Cara-cara yang terorganisir dalam mengumpulkan, mengolah, mengendalikan, menyimpan dan melaporkan informasi untuk mencapai suatu tujuan itulah sistem informasi (Nova et al., 2022).

Dengan adanya pemanfaatan sistem informasi yang benar, maka dapat meningkatkan kinerja serta menunjang kegiatan yang dilakukan dan dapat menjadi suatu keunggulan untuk menghadapi persaingan (Ashari et al., 2022).

Dari uraian-uraian yang ada diatas, maka penulis menyimpulkan bahwa sistem informasi merupakan suatu sistem yang terintegrasi satu sama lainnya dengan menggunakan teknologi informasi yang saling bekerja sama dalam mengumpulkan, mengolah, mengendalikan, menyimpan dan mendistribusi informasi untuk mendukung suatu kegiatan dalam suatu lingkup tertentu serta memberikan laporan-laporan yang relevan.

2.1.4 Perancangan

Pengertian perancangan menurut para ahli yaitu sebuah proses yang menggunakan berbagai macam prinsip dan teknologi untuk merealisasikan bentuk fisik dari pendefinisian suatu perangkat, proses ataupun sistem sebagai tingkat detail tertentu (Elisa et al., 2020).

2.1.5 SDLC

Software Development Life Cycle atau *SDLC* adalah sebuah metodologi yang digunakan dalam proses pengembangan sistem komputer atau sistem informasi yang memiliki tahapan-tahapan terstruktur dimulai dari perencanaan, analisa, desain, implementasi, pengujian dan perawatan (Ridwan et al., 2021).

2.1.6 Agile

Agile adalah salah satu pendekatan dalam *Software Development Life Cycle* (*SDLC*) yang berfokus pada orang, asosiasinya terhadap satu sama lain, perangkat lunak yang berfungsi, koalisi pelanggan dan perubahan, dibanding pada metode, alat, kontrak dan rencana (Hayat et al., 2019). Metode ini mendukung perencanaan yang adaptif, perkembangan evolusioner, pengiriman awal, peningkatan terus-menerus dan perubahan akan kebutuhan yang fleksibel disetiap saat dari proses pengembangan perangkat lunak (Mallidi & Sharma, 2021).

Metode *agile* ini terdiri dari beberapa model yang berbeda-beda untuk mendukung berbagai *Software Development Life Cycle* (*SDLC*) yang ada (Mallidi & Sharma, 2021), model-model tersebut seperti *Agile Unified Process* (*AUP*), *Extreme Programming* (*XP*), *Adaptive Software Development* (*ASD*), *Feature-Driven Development* (*FDD*), *Kanban*, *Scrum* dan *Scaled Agile Framework* (*SAFe*).

Masing-masing model *agile* memiliki fungsi dan cara kerja yang sedikit berbeda dari yang lainnya, namun semuanya masih tetap menggunakan prinsip dalam metode *agile*, seperti model *kanban* yang digunakan untuk melacak kemajuan dan divisualisasikan menggunakan papan *kanban*, sehingga

menghasilkan transparansi pada setiap pekerjaan anggota tim (Purwanto et al., 2022) dan model *scrum* yang merupakan model berbasis iteratif dan inkremental yang membangun perangkat lunak dengan mekanisme yang telah ditentukan, seperti halnya modul perangkat lunak yang dapat berkembang dalam potongan kecil dengan cara yang berulang (Hayat et al., 2019).

Meskipun terdapat berbagai model dan bentuk dalam metode *agile*, semuanya memiliki tujuan yang sama, yaitu mengirimkan perangkat lunak yang berfungsi kepada pengguna dengan tepat waktu.

2.1.7 UML

Unified Modeling Language atau *UML* pertama kali diperkenalkan pada tahun 1997, kini telah berkembang menjadi bahasa pemodelan yang baku dalam sebuah pengembangan perangkat lunak, terutama dalam pengembangan yang menggunakan pendekatan berorientasi objek (T. A. Kurniawan, 2018).

Menurut (Elisa et al., 2020), *UML* merupakan suatu bahasa pemodelan bagi sistem atau perangkat lunak yang berorientasi pada objek, yang berfungsi untuk memudahkan pemahaman atas masalah-masalah yang kompleks. Menurut (Sovia et al., 2020), beberapa diagram *UML* yang umum digunakan terdiri dari beberapa jenis sebagai berikut :

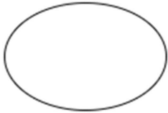

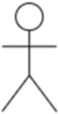

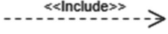
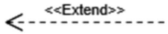
1. *Use case diagram*

Diagram ini mendeskripsikan interaksi yang terjadi antara pengguna sistem dalam suatu lingkup dengan sebuah cerita tentang bagaimana sistem tersebut digunakan, yang terdiri dari aktor, yaitu manusia, perangkat keras atau sistem, dan





interaksi-interaksi yang dilakukannya (T. B. Kurniawan & Syarifuddin, 2020).

Notasi yang digunakan dalam *use case* diagram dapat dilihat pada Tabel 2.1.

Tabel 2.1 Notasi *use case* diagram

Gambar	Nama	Keterangan
	<i>Use case</i>	<i>Use case</i> mewakili tujuan pengguna yang dapat dicapai dengan mengakses sistem atau aplikasi perangkat lunak.
	<i>Association</i>	<i>Association</i> berhubungan dengan urutan tindakan antara aktor dan <i>use case</i> dalam mencapai <i>use case</i>
	<i>Actor</i>	<i>Actor</i> adalah entitas yang berinteraksi dengan sistem
	<i>System</i>	<i>System</i> merepresentasikan ruang lingkup atau batas sistem.
	<i>Include</i>	<i>Include</i> menentukan bagaimana perilaku untuk kasus penggunaan inklusi dimasukkan ke dalam perilaku yang ditentukan untuk kasus penggunaan dasar.
	<i>Extend</i>	<i>Extend</i> menentukan bagaimana perilaku kasus penggunaan ekstensi dapat dimasukkan ke dalam perilaku yang ditentukan untuk kasus penggunaan dasar.

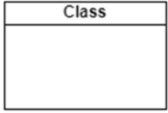






Tabel 2.1 Lanjutan

	<i>Dependency</i>	<i>Dependency</i> menyatakan bahwa elemen model bergantung pada elemen model lain untuk spesifikasi dan/atau implementasi.
	<i>Generalization</i>	<i>Generalization</i> digunakan untuk merepresentasikan hubungan pewarisan antara elemen model dengan tipe yang sama.
	<i>Realization</i>	<i>Realization</i> adalah hubungan antara spesifikasi dan implementasinya.
	<i>Collaboration</i>	<i>Collaboration</i> menggambarkan struktur elemen (peran) yang berkolaborasi, masing-masing melakukan fungsi khusus, yang secara kolektif mencapai beberapa fungsi yang diinginkan.




2. Class diagram

Diagram ini menggambarkan keadaan dari suatu sistem, memberikan layanan untuk memanipulasi keadaan tersebut dan juga menggambarkan struktur, deskripsi *class*, *package*, *object*, serta hubungannya antara satu dengan lainnya seperti pewarisan (*inheritance*), asosiasi (*association*) dan lain-lain (S. Wahyudi, 2020). Notasi yang digunakan dalam class diagram dapat dilihat pada Tabel 2.2.

Tabel 2.2 Notasi *class* diagram

Gambar	Nama	Keterangan
	<i>Class</i>	<i>Class</i> mewakili sekumpulan objek yang memiliki struktur, perilaku, dan hubungan yang sama dengan objek <i>class</i> lain.
	<i>Generalization</i>	<i>Generalization</i> adalah asosiasi antara <i>classifier</i> yang lebih umum dan <i>classifier</i> yang lebih khusus.
	<i>Realization</i>	<i>Realization</i> adalah hubungan antara <i>interface</i> dan <i>class</i> atau komponen yang mewujudkannya.
	<i>Association</i>	<i>Association</i> adalah hubungan yang menghubungkan dua <i>class</i> .
	<i>Aggregation</i>	<i>Aggregation</i> adalah asosiasi dengan hubungan antara keseluruhan dan bagian-bagiannya, relasi ketika satu <i>class</i> adalah entitas tertentu yang menyertakan entitas lain sebagai komponen.
	<i>Composition</i>	<i>Composition</i> adalah varian <i>aggregation</i> yang kuat ketika bagian-bagian tidak dapat dipisahkan dari keseluruhan.
	<i>Dependency</i>	<i>Dependency</i> adalah hubungan ketika beberapa perubahan dari satu elemen model dapat membutuhkan perubahan elemen dependen lainnya.


Tabel 2.2 Lanjutan

	<i>N-ary Association</i>	<i>N-ary association</i> mewakili dua atau lebih <i>aggregation</i> .
	<i>Collaboration</i>	<i>Collaboration</i> mendeskripsikan urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>Package</i>	<i>Package</i> mengelompokkan <i>class</i> dan <i>package</i> lainnya.







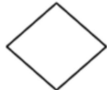
3. *Activity* diagram

Diagram ini menggambarkan aktivitas atau alur kerja sebuah sistem atau sebuah proses bisnis (Elisa et al., 2020). Diagram ini juga menggambarkan berbagai alir dari aktivitas di dalam sistem yang sedang dirancang seperti bagaimana alir berawal, berakhir dan keputusan-keputusan yang mungkin terjadi diantaranya (S. Wahyudi, 2020). Notasi yang digunakan dalam *activity* diagram dapat dilihat pada Tabel 2.3.

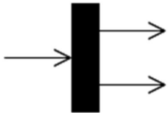
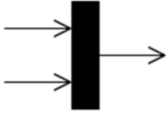
Tabel 2.3 Notasi *activity* diagram

Gambar	Nama	Keterangan
	<i>Initial Node</i>	<i>Initial node</i> digunakan untuk mewakili titik awal atau keadaan awal dari suatu aktivitas.

Tabel 2.3 Lanjutan

	<p><i>Activity</i></p>	<p><i>Activity</i> digunakan untuk mewakili proses aktivitas.</p>
	<p><i>Action</i></p>	<p><i>Action</i> digunakan untuk mewakili sub-area yang dapat dieksekusi dari suatu aktivitas.</p>
	<p><i>Control Flow</i></p>	<p><i>Control flow</i> digunakan untuk mewakili aliran kontrol dari satu aksi ke aksi lainnya.</p>
	<p><i>Object Flow</i></p>	<p><i>Object flow</i> digunakan untuk mewakili jalur objek yang bergerak melalui aktivitas.</p>
	<p><i>Activity Final Node</i></p>	<p><i>Activity final node</i> digunakan untuk menandai akhir dari semua alur kontrol dalam aktivitas.</p>
	<p><i>Flow Final Node</i></p>	<p><i>Flow final node</i> digunakan untuk menandai akhir dari aliran kontrol tunggal.</p>
	<p><i>Decision Node</i></p>	<p><i>Decision node</i> digunakan untuk mewakili titik cabang kondisional dengan satu <i>input</i> dan beberapa <i>output</i>.</p>


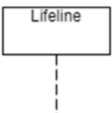
Tabel 2.3 Lanjutan

	<p><i>Fork Node</i></p>	<p><i>Fork node</i> digunakan untuk mewakili aliran yang mungkin bercabang menjadi dua atau lebih aliran paralel.</p>
	<p><i>Join Node</i></p>	<p><i>Join node</i> digunakan untuk mewakili dua <i>input</i> yang bergabung menjadi satu <i>output</i>.</p>

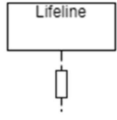

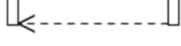


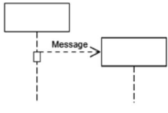
4. *Sequence diagram*

Diagram ini menunjukkan interaksi antar objek di dalam sistem yang tersusun berdasarkan urutan atau rangkaian waktu, yang digunakan untuk menggambarkan serangkaian langkah sebagai suatu tanggapan atas kejadian untuk menghasilkan sebuah keluaran (*output*) (S. Wahyudi, 2020). Notasi yang digunakan dalam *sequence diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Notasi *sequence diagram*

Gambar	Nama	Keterangan
	<p><i>Actor</i></p>	<p><i>Actor</i> mewakili peran yang dimainkan oleh pengguna manusia, perangkat keras eksternal, atau subjek lainnya.</p>
	<p><i>Lifeline</i></p>	<p><i>Lifeline</i> mewakili peserta individu dalam Interaksi.</p>

Tabel 2.4 Lanjutan

	<p><i>Activation</i></p>	<p><i>Activation</i> merupakan persegi panjang tipis pada <i>lifeline</i> yang mewakili periode selama elemen melakukan operasi.</p>
	<p><i>Call Message</i></p>	<p><i>Call message</i> adalah jenis <i>message</i> yang mewakili permintaan operasi pada <i>lifeline</i> target.</p>
	<p><i>Return Message</i></p>	<p><i>Return message</i> adalah jenis <i>message</i> yang mewakili penyampaian informasi kembali ke <i>caller</i> dari pesan koresponden sebelumnya.</p>
	<p><i>Self Message</i></p>	<p><i>Self message</i> adalah jenis <i>message</i> yang mewakili permintaan <i>message</i> dari <i>lifeline</i> yang sama.</p>
	<p><i>Recursive Message</i></p>	<p><i>Recursive message</i> adalah jenis <i>message</i> yang mewakili permintaan <i>message</i> dari <i>lifeline</i> yang sama. Targetnya menunjuk ke aktivasi di atas aktivasi tempat <i>message</i> itu dipanggil.</p>
	<p><i>Create Message</i></p>	<p><i>Create message</i> adalah jenis <i>message</i> yang merepresentasikan instantiasi dari (target) <i>lifeline</i>.</p>

2.1.8 Mikrokontroler

Mikrokontroler merupakan sebuah *single chip microcomputer*, dimana sebagian besar atau seluruh elemen-elemennya dirangkum di dalam satu *chip IC*

(Septiadi & Alfarizi, 2020). Berbeda dengan mikroprosesor, mikrokontroler secara umum terdapat komponen pendukung sistem minimal mikroprosesor yang dapat diprogram untuk melakukan tugas spesifik dan dapat menghasilkan sebuah keluaran atau *output* (Septryanti et al., 2022).

2.1.9 XAMPP

XAMPP adalah sebuah perangkat lunak gratis (*freeware*), yang dikembangkan untuk dapat dioperasikan diberbagai macam sistem operasi dan memiliki fungsi sebagai *server* yang berdiri sendiri (*stand-alone*) yang dikenal dengan sebutan *Localhost* (Dalimunthe, 2020). *XAMPP* dirancang sedemikian rupa untuk dapat digunakan diberbagai macam sistem operasi dengan fungsi untuk menampilkan perintah *PHP* yang telah dirancang pemrogram dan data yang telah disimpan dalam *MySQL* di penyimpanan lokal (Lesmana & Silalahi, 2022).

2.1.10 Database

Secara umum, *database* merupakan kumpulan dari *file* yang terkait serta berinteraksi dengan kunci *file* yang ada (Asshiddiqi et al., 2022), jadi *database* dapat diartikan sebagai media pengumpulan data yang saling terkait dan terorganisasi untuk penggunaan kembali secara cepat dan mudah (Brawijaya et al., 2020). Dengan kata lain, *database* itu digunakan tidak hanya untuk menyimpan data, tetapi juga untuk mendapatkan informasi dan pengetahuan (Dalimunthe, 2020).

2.1.11 *MySQL*

MySQL adalah perangkat lunak *server database* yang memiliki kemampuan untuk memproses pengiriman dan penerimaan data dengan cepat serta *multi user* (Dalimunthe, 2020). *MySQL* pertama kali ditemukan oleh Michael Widenius di tahun 1979, dimana *MySQL* ini digunakan sebagai *database website* yang bisa diintegrasikan dengan bahasa pemrograman, berfungsi dalam *hardware* dengan spesifikasi rendah serta memiliki standar industri (Krismanto et al., 2022). *MySQL* menggunakan *Structured Query Language (SQL)* sebagai bahasa standar untuk menyimpan, memanipulasi dan mengambil data dalam sebuah *database* (Destriana et al., 2020).

2.1.12 *PHP*

PHP adalah sebuah akronim rekursif untuk *PHP: Hypertext Preprocessor*, yang merupakan sebuah perangkat lunak *open source* berbasis bahasa pemrograman *server-side scripting*, dimana seluruh *script* dari *PHP* akan diproses seluruhnya oleh bagian *server* (Purwanto et al., 2022). Menurut (Destriana et al., 2020), ia juga berpendapat bahwa *PHP* merupakan suatu *bahasa server-side scripting*, yaitu seluruh sintaks dan perintah yang ada akan sepenuhnya dijalankan dalam sistem, yang menyatu dengan *HTML* untuk membuat sebuah halaman *website* yang dinamis.

2.2 Tinjauan Teori Khusus

Di dalam sub bab berikut, penulis akan membahas mengenai teori-teori

khusus yang berhubungan dengan pelaksanaan penelitian ini yang berupa pengertian-pengertian secara umum.

2.2.1 Absensi

Absensi merupakan suatu proses dimana seorang individu melakukan pembuktian terhadap kehadirannya disuatu tempat yang dicatat disuatu media tertentu (Mulia, 2020). Absensi dapat dijadikan sebagai suatu indikator penilaian kinerja individu atau kelompok dalam melakukan tugasnya (Helling & Apriyani, 2020).

2.2.2 *Internet of Things*

Internet of Things merupakan suatu konsep yang menyatakan bahwa perangkat-perangkat yang ada dan terhubung dengan jaringan internet mampu berkomunikasi dan bertukar informasi (Puryono & Handayani, 2021). Tujuan dari *Internet of Things* yaitu mengaktifkan berbagai fungsi seperti identifikasi, lokasi, pelacakan, pemantauan dan manajemen dalam berbagai konteks seperti privasi, industri, kesehatan dan lain-lain (Duroc, 2022).

2.2.3 *RFID*

RFID adalah sebuah akronim untuk *Radio-Frequency Identification*, yang mengacu pada teknologi di mana data digital yang dikodekan dalam *tag RFID* atau *smart label* ditangkap oleh pembaca melalui gelombang radio untuk mengidentifikasi orang ataupun objek secara otomatis (Hu et al., 2020). RFID

memiliki kemiripan dengan teknologi *barcode* dimana teknologi *barcode* membaca kode dalam bentuk garis berbaris. Hanya saja *tag RFID* dapat menyimpan lebih banyak data jika dibandingkan dengan teknologi *barcode* (Juhana, 2021). Selain itu, *RFID* juga tidak perlu membutuhkan kontak langsung ataupun garis pandang untuk berkomunikasi (Hu et al., 2020).

RFID secara umum terdiri dari dua komponen utama yaitu *RFID tag* dan *RFID reader*. Setiap *RFID tag* memiliki nomor seri yang berbeda yang ditanamkan pada *IC* memori untuk menyimpan informasi yang diperlukan. Sedangkan *RFID reader* berfungsi sebagai pembaca informasi yang terdapat dalam *RFID tag* (Septiadi & Alfarizi, 2020).

Menurut Lempert dan Pflaum (2011) yang disebutkan dalam (Chocholac et al., 2021), teknologi identifikasi otomatis seperti *RFID* memiliki fungsi sebagai berikut : identifikasi (identifikasi unik objek), lokasi (informasi akurat tentang posisi objek), sensor (status objek saat ini), komunikasi (aksesibilitas objek), penyimpanan data (retensi riwayat objek), dan logika (pengenalan peristiwa kritis objek).

2.2.4 ESP32

ESP32 merupakan mikrokontroler yang dikembangkan oleh *Espressif System*, yang merupakan penerus dari mikrokontroler *ESP8266*. *ESP32* ini mendukung pembuatan aplikasi *Internet of Things*, karena pada mikrokontroler ini tersedia modul *WiFi* di dalam *chipnya*. *ESP32* sendiri tidak jauh berbeda dengan *ESP8266* yang terkenal di pasaran, hanya saja *ESP32* lebih kompleks dibandingkan

ESP8266, dan lebih banyak digunakan untuk proyek yang besar (Muliadi et al., 2020).

Mikrokontroler *ESP32* ini dapat diprogram dengan menggunakan C++, C, Python, Lua dan lain-lain. Untuk menjalankan program mikrokontroler *ESP32* ini memerlukan suatu software pemrograman seperti Arduino Promini, Arduino IDE, Ubuntu 14.04 LTS, ESP-IDF Visual Studio Code Extension, Espressif IoT Development Framework dan lain sebagainya.

2.2.5 Web

World Wide Web atau lebih sering dikenal dengan sebutan web atau *website* merupakan suatu layanan penyajian informasi yang menggunakan konsep *hyperlink* (tautan) (F. Wahyudi, 2020). *Website* atau *web* merupakan kumpulan dari halaman-halaman yang saling berhubungan dan memiliki informasi data digital berupa teks, gambar, video dan lain sebagainya yang dapat diakses melalui sebuah jaringan internet (Brawijaya et al., 2020). *Website* berguna untuk menampilkan berbagai informasi dalam bentuk teks, gambar, video dan lain sebagainya (Elisa et al., 2020).