

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 Pengertian Sistem

Sistem adalah kumpulan dari komponen yang saling berhubungan. Sistem memiliki banyak karakteristik, seperti: (Hutahaean, 2014)

1. **Komponen**

Setiap sistem terdiri dari berbagai bagian yang berinteraksi satu sama lain untuk membentuk satu kesatuan..

2. **Batasan sistem (*Boundary*)**

Batasan sistem adalah area yang membatasi suatu sistem dengan sistem lain atau lingkungan eksternal..

3. **Lingkungan Luar Sistem (*Environment*)**

Lingkungan di luar sistem memengaruhi operasi sistem..

4. **Penghubung Sistem (*Interface*)**

Penghubung sistem berfungsi untuk menghubungkan subsistem satu sama lain..

5. **Masukkan Sistem (*Input*)**

Energi yang dimasukkan ke dalam sistem disebut masukkan. Ini dapat berupa input perawatan atau input sinyal.

6. **Keluaran Sistem (*Output*)**

Energi yang diproses dibagi menjadi keluaran yang berguna dan sisa pembuangan..

7. **Pengolah Sistem**

Suatu sistem berubah menjadi komponen pengolah yang mengubah masukkan menjadi keluaran..

8. **Sasaran Sistem**

Setiap sistem pasti memiliki tujuan atau sasaran.

2.1.2 Pengertian Informasi

Informasi adalah data yang telah diubah menjadi bentuk yang bermanfaat bagi orang yang menerimanya dan memiliki nilai yang dapat dilihat atau digunakan dalam membuat keputusan saat ini atau di masa depan. (Hutahaean, 2014)

2.1.3 Pengertian Sistem Informasi

Sistem informasi adalah proses pengumpulan, penyimpanan, dan analisis informasi untuk tujuan tertentu. Sistem informasi terdiri dari data (*input*) dan laporan (*output*), yang digunakan untuk melakukan kegiatan strategis dan diterima oleh sistem lainnya. (Prehanto, 2020)

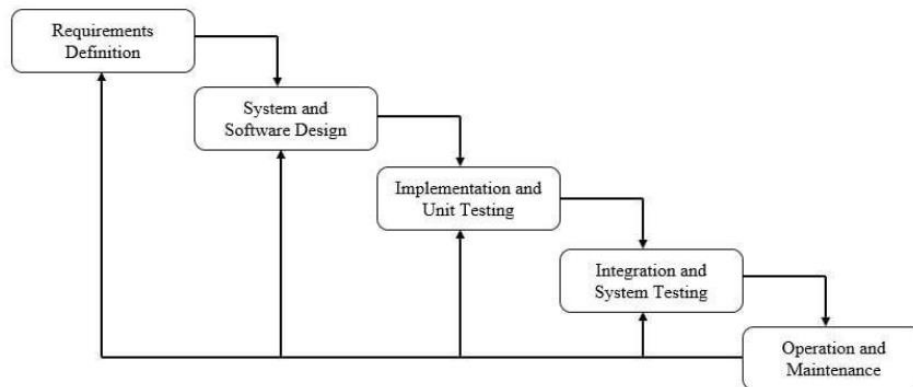
2.1.4 Pengertian Administrasi

Perencanaan, pengendalian, dan pengorganisasian operasi perkantoran serta penggerak mereka untuk melaksanakannya untuk mencapai tujuan yang telah ditetapkan dikenal sebagai administrasi. (Taufiqrokhman et al., 2021).

2.1.5 *System Development Life Cycle (SDLC)*

Software Development Life Cycle (Siklus Hidup Pengembangan Perangkat Lunak), juga dikenal sebagai SDLC, adalah proses pengembangan atau penyesuaian sistem perangkat lunak menggunakan metodologi dan model yang sama yang digunakan orang untuk mengembangkan sistem perangkat lunak sebelumnya. (A. S. & Shalahuddin, 2013).

Lima tahapan metode *waterfall* menurut Ian Sommerville (2011) digambarkan pada **Gambar 2.2** di bawah ini: (Herlina et al., 2022)



Gambar 2.1 SDLC Waterfall

(Sumber : Herlina Rusman, Putri Dwi Ayu Suwardoyo, Untung, 2022)

Berikut adalah penjelasan dari tahapan-tahapan tersebut:

1. *Requirement Analysis and Definition*

Proses ini mencakup diskusi dengan pengguna sistem untuk menentukan fitur, hambatan, dan tujuan sistem. Hal ini akan ditetapkan secara menyeluruh dan memiliki fungsi spesifikasi sistem..

2. *System and Software Design*

Pada tahap ini dilakukan evaluasi informasi tentang spesifikasi kebutuhan dari tahap analisis kebutuhan. Informasi ini kemudian diterapkan pada desain pengembangan. Perencanaan desain dilakukan untuk membantu memberikan gambaran keseluruhan pekerjaan yang akan dilakukan. Selain itu, tahapan ini akan membantu pengembang mempersiapkan kebutuhan perangkat keras selama proses keseluruhan pembuatan arsitektur sistem yang akan dibangun..

3. *Implementation and Unit Testing*

Tahap implementasi dan pengujian unit adalah tahapan pemrograman. Tahap pembuatan sistem lunak terbagi beberapa modul yang lebih kecil yang nantinya akan digabungkan. Selain itu, pada tahap ini pengujian dan

pengecekan untuk memastikan bahwa modul yang telah dibuat memenuhi kriteria yang diinginkan..

4. *Integration and System Testing*

Setelah semua unit atau modul dibuat dan diuji pada tahap implementasi, kemudian dibutuhkan integrasi ke sistem secara menyeluruh. Selesai melakukan proses integrasi, sistem secara keseluruhan diperiksa dan diuji untuk kesalahan dan kegagalan..

5. *Operation and Maintenance*

Perangkat lunak siap pakai dioperasikan dan dirawat oleh pengguna pada tahap terakhir Metode Waterfall. Pemeliharaan mampu membantu pengembang memperbaiki kesalahan yang tidak terdeteksi. Ini termasuk perbaikan kesalahan, perbaikan implementasi unit, dan peningkatan serta penyesuaian sistem.

2.1.7 **OOAD**

Cara berpikir baru tentang masalah menggunakan model yang dibangun di atas konsep dunia nyata dikenal sebagai analisis dan desain berorientasi objek..

2.1.8 ***Unified Modelling Language (UML)***




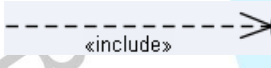
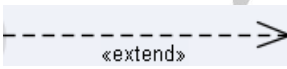
Unified Modelling Language adalah metode pemodelan visual yang digunakan untuk membangun sistem berorientasi objek. Untuk pertama kalinya, UML dibuat sebagai bahasa standar penulisan blueprint software atau bahasa standar, merancang, dan dokumentasi sistem. Diharapkan UML dapat memudahkan RPL dan memenuhi semua kebutuhan pengguna dengan cara yang efisien, lengkap, dan tepat. Faktor-faktor seperti scalability, robustness, dan security termasuk dalam kategori ini..

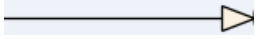
2.1.9 ***Use Case Diagram***

Salah satu jenis diagram digunakan untuk menunjukkan hubungan antara sistem dan aktor adalah use case diagram. Jenis interaksi apa

pun yang dilakukan oleh pengguna sistem dengan sistem dapat dijelaskan melalui use case diagram. Use case diagram mudah dipelajari. Untuk memulai pemodelan, Anda harus membuat diagram yang dapat menggambarkan tindakan aktor dengan tindakan sistem, seperti yang dilakukan pada diagram Use Case. Di bawah ini, **Tabel 2.1** menunjukkan simbol-simbol yang digunakan untuk masing-masing kasus..

Tabel 2.1 Simbol Use Case

Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan tugas individu, sistem, atau alat komunikasi dengan use case
	<i>Use Case</i>	Interaksi antara aktor dan sistem
	<i>Assosiation</i>	Detail tentang hubungan aktor dengan use case
	<i>Include</i>	Menunjukkan bahwa fungsionalitas dari suatu use case adalah fungsi use case yang lain
	<i>Exclude</i>	menunjukkan bahwa jika suatu kondisi terpenuhi, suatu use case dapat ditambahkan secara fungsional ke use case lainnya.






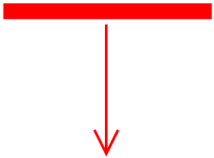
	<i>generalization</i>	Menunjukkan spesialisasi aktor untuk melakukan use case
---	-----------------------	---

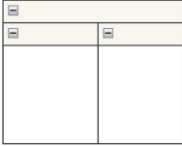
(Sumber : (Redaksi, 2022))

2.1.10 Activity Diagram

Activity diagram adalah versi tambahan dari *Use Case* yang menampilkan alur aktivitas. Aktivitas berupa rangkaian menu yang dilakukan sistem. "Activity diagram digunakan untuk menganalisa diagram use case dengan memberi tahu aktor apa yang harus dilakukan dan kapan harus dilakukan.. Di bawah ini, **Tabel 2.2** menunjukkan simbol yang digunakan pada *activity* diagram.

Tabel 2. 2 Simbol Activity

Simbol	Nama	Keterangan
	<i>Initial Node</i>	Status awal ditunjukkan pada diagram aktivitas.
	<i>Activity</i>	Kebanyakan aktivitas mampu dikerjakan sistem dimulai dengan kata kerja.
	<i>Decision</i>	Percabangan adalah ketika Anda dapat melakukan lebih dari satu hal.
	<i>Line Connector</i>	hubungan antara aktivitas.
	<i>Final Node</i>	Status akhir sistem.
	<i>Fork</i>	Penggabungan lebih dari satu pekerjaan


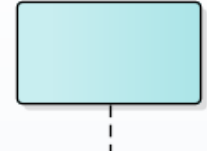
		digabungkan menjadi satu.
	<i>Swim Lane</i>	Memisahkan antar organisasi atas aktivitas yang terjadi.



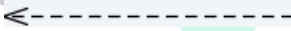
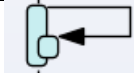

(Sumber : Intern, 2021))

2.1.11 Sequence Diagram

Sequence diagram, juga dikenal sebagai diagram urutan, adalah diagram yang memiliki fungsi untuk menunjukkan interaksi antar objek-objek dalam sebuah sistem secara menyeluruh. *Sequence diagram* juga menunjukkan pesan atau perintah yang dikirim, serta waktu yang diperlukan untuk menyelesaikannya. Biasanya, barang-barang yang berkaitan dengan proses operasi terurut dari kiri menuju kanan. Diagram ini memiliki dua dimensi, dimensi vertikal menunjukkan waktu, dan dimensi horizontal menunjukkan objek. Kolom vertikal atau disebut *lifeline* menunjukkan waktu aktif tiap objek, termasuk aktor. Untuk saat ini, perintah atau pesan ditunjukkan sebagai garis panah yang menghubungkan satu *lifeline* dengan *lifeline* lain. Di bawah ini, **Tabel 2.3** menunjukkan simbol-simbol yang digunakan dalam rangkaian..

Tabel 2. 3 Simbol-Simbol Sequence

Simbol	Nama	Keterangan
	<i>Actor</i>	mengkomunikasikan entitas yang tidak termasuk dalam sistem dan berinteraksi dengannya.
	<i>LifeLine</i>	menghubungkan objek selama rangkaian (pesan diterima dan aktivasi).

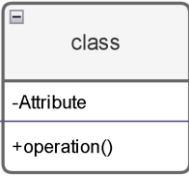




	<i>Fragment</i>	Berfungsi untuk yang membungkus atau menyatukan beberapa bagian interaksi pada <i>sequence</i> diagram dalam kotak.
	<i>Message Entry</i>	Berfungsi untuk menunjukkan pesan antar objek dengan urutan kejadian yang terjadi.
	<i>Message to Self</i>	menggambarkan pesan objek yang mengarahkan urutan peristiwa yang terjadi.
	<i>Recursive</i>	menggambarkan hasil pengiriman pesan dan digambarkan dari kanan ke kiri.
	<i>A Focus of Control (Activation)</i>	Mewakili proses durasi aktivasi sebuah operasi.

(Sumber : (Ansori, 2022))

2.1.12 Class Diagram

Salah satu jenis diagram struktur yang tersedia dalam UML adalah diagram *class*, yang menampilkan struktur, deskripsi class, atribut, metode, dan hubungan setiap objek. Namun, diagram *class* tidak menjelaskan maksud pada kelas, tetapi hanya menunjukkan hubungan yang terjadi. Di bawah ini adalah contoh simbol pada *class* diagram, yang dapat dilihat pada **Tabel 2.4**.

Tabel 2. 4 Simbol-Simbol Class

Simbol	Nama	Keterangan
	<i>Class</i>	Bisa memetakan himpunan objek dengan atribut dan operasi yang sama.
	<i>Association</i>	memetakan hubungan antara objek.
	<i>Generalization</i>	menunjukkan hubungan objek yaitu perilaku dan data struktur objek induk.
	<i>Agregation</i>	Kelas antara kelas yang memiliki arti setiap bagian
	<i>Composition</i>	jenis agresi khusus kelas yang membentuk whole.
	<i>Dependency</i>	Ini berarti menghubungkan hubungan apabila perubahan pada satu elemen mandiri berdampak pada elemen lain yang tidak berdiri sendiri.

(Sumber : (Yahya, 2022))

2.1.13 Black Box

Black Box testing merupakan pengujian perangkat lunak. Tujuan pengujian ini adalah untuk menentukan apakah perangkat lunak dapat berfungsi dengan baik setelah dibuat. Penguji tidak perlu menulis kode program agar pengujian dapat dilakukan. Tes ini dapat dilakukan oleh siapa saja..

2.2 Tinjauan Studi

Review digunakan untuk membantu dan mendukung penelitian yang dilakukan:

1. Jurnal Teknik Informatika, Volume 10, No. 3 (2018), dibuat oleh I Made Dharma, I Ngurah Gede Wiweka Naren Ananda, dan Mochamad Rizki Alzaki yang memiliki judul : Rancang Bangun Aplikasi Pemesanan Jasa Service Komputer Berbasis Web. Jurnal ini membahas mengenai perancangan dan implementasi sistem layanan jasa *service* komputer untuk pelanggan. Permasalahan yang dibahas oleh penulis yaitu pelanggan menginginkan jasa layanan *service* komputer via ticket, informasi status *service* , dan kurangnya pencatatan data riwayat *service*. Hasil yang didapat dari rancangan ini adalah sistem pelayanan, informasi status dan pendataan riwayat *service* berjalan dengan sempurna. Berdasarkan penelitian ini , informasi yang diberikan oleh jurnal berguna untuk menambahkan penelitian riset saya mengenai reservasi *online* via *website* jasa *service* komputer (Dharma et al., 2018).
2. Jurnal Prosiding Seminar Nasional Informatika dan Sistem Informasi Volume 5, No. 1 (2020), Judul : Sistem Informasi Reservasi Penjadwalan Service Pada PT. Mentari Alam Semesta Berbasis Web. Jurnal ini membicarakan desain sistem reservasi konsumen. Permasalahan yang dibahas oleh penulis termasuk kesulitan bagi pelanggan untuk mendapatkan nomor antrian, kesulitan bagi pelanggan untuk mendapatkan informasi tentang biaya dan harga sparepart di bengkel, dan kesulitan bagi pemilik bengkel untuk mengelola data servis setiap hari. Hasil dari desain sistem reservasi adalah bahwa sistem tersebut dapat menyediakan layanan servis dan membuat manajemen data bengkel lebih mudah. Berdasarkan penelitian ini, informasi ini membantu saya menyempurnakan penelitian penelitian saya tentang reservasi online

melalui situs web penyedia layanan komputer (Dadang Kurnia dan Hendri Ardiansyah, 2020).

3. Jurnal PHASTI, Volume 6, No. 1 (2021), dibuat oleh Muhammad Ichsan dan Fitria yang memiliki judul : Sistem Informasi Service Komputer Berbasis Web Studi Kasus DataCom. Pembahasan dalam jurnal mengenai desain sistem layanan komputer yang terkomputerisasi berbasis webiste. Hasil yang di dapat bahwa sistem ini dirancang untuk mempermudah teknisi dan admin servis untuk menyimpan data servis komputer tanpa menghabiskan banyak ruang penyimpanan. Menu laporan sistem memiliki fitur cetak, yang memungkinkan Anda kelola data bulanan dalam format PDF. Hasil jurnal ini dapat saya gunakan dalam penelitian saya tentang toko Makmur yang membuat sistem reservasi jasa service; pendataan reservasi dapat ditata dan dikelola dengan baik. Hasil dari jurnal ini yang dapat saya ambil untuk di aplikasikan kedalam riset saya mengenai toko Makmur yang membuat sistem reservasi jasa *service* adalah pendataan dari reservasi mampu ditata atau dikelola dengan rapi (Ichsan & Fitria, 2021).
4. Jurnal DEWANTARA. J. Tech., Volume 2, No. 2 (2021), dibuat oleh Ummul Khair, Ahmad Ali Hakam Dani, dan Muhlis Muhallim yang memiliki judul : Sistem Informasi Pengelolaan Service Komputer Berbasis Web Di Central Media Computer Palopo. Jurnal ini membahas pengelolaan dan implementasi layanan komputer berbasis website. Dan juga, pengolahan data khususnya penerimaan barang layanan saat ini pada central media computer Palopo, belum terkomputerisasi dan berbasis web. Hasil akhir dari pengelolaan dan implementasi ini diharapkan dapat membantu dan memudahkan layanan komputer dan memberikan layanan layanan secara cepat. Hasil yang dapat saya ambil mengenai jurnal ini adalah riset saya yang berfokus pada reservasi mampu membantu pendataan untuk

admin toko untuk memberikan pelayanan *service* yang mudah dan cepat kepada pelanggan (Khair M- et al., 2021).

5. Jurnal INSAN (Journal of Information System Management Innovation) Volume 1, No. 1 (2021), dibuat oleh Fernando B. Siahaan, Bismar Jati Sakti, Khairul Anwar, Muhammad Baldin Fajrind, dan Riswandi Ishak yang memiliki judul : Perancangan Sistem Informasi Penanganan Service Komputer Berbasis Web (Sirespuwan). Jurnal ini membahas tentang proses penanganan yang lambat dan waktu yang dibutuhkan untuk menyelesaikannya. Ini disebabkan oleh fakta bahwa proses penanganan komputer di PT Askara Internal masih dilakukan secara manual. Penulis kemudian merancang sebuah sistem informasi web yang dapat mengatasi lambatnya proses penanganan data pelanggan dengan mengisikan form digital. Hasil jurnal ini menunjukkan bahwa sistem informasi web ini sangat membantu dalam menangani kerusakan komputer karena kerusakan dapat ditangani dengan cepat dan teknisi dapat segera mengetahui tentang keluhan atau komplain. Hal ini sangat membantu pengguna dalam proses komplain karena ada respons yang sangat cepat untuk keluhan di internet.. Dari hasil jurnal diatas , informasi yang saya dapatkan akan membantu dalam riset saya mengenai reservasi jasa *service* komputer berbasis *website*. Pendataan yang akan di dapat dari pelanggan akan sangat berpengaruh kedalam pengelolaan data admin. Oleh karenanya pembuatan form berbasis *website* akan mempermudah untuk mengelolanya (Siahaan et al., 2021).