

BAB II

TINJAUAN PUSTAKA

2.1. Teori Dasar

Pada laporan ini, penulis melakukan studi terhadap berbagai teori yang akan dijadikan acuan, antara lain sebagai berikut:

2.1.1 Manajemen Aset

Manajemen aset (Asset Management) merupakan suatu pendekatan strategis yang melibatkan pengelolaan seluruh siklus hidup aset perusahaan dengan tujuan untuk meningkatkan nilai aset dan efisiensi operasional. Konsep ini mencakup identifikasi, akuisisi, penggunaan, pemeliharaan, dan penarikan aset. Manajemen aset tidak hanya fokus pada aset fisik seperti properti, tanah, dan peralatan, tetapi juga dapat mencakup aset tak berwujud seperti merek dan hak paten. Sumber teorinya dapat ditemukan dalam berbagai literatur dan pedoman manajemen aset, di antaranya "ISO 55000:2014 Asset Management - Overview, Principles, and Terminology." ISO 55000 adalah standar internasional yang memberikan panduan mengenai prinsip-prinsip dasar dan terminologi dalam manajemen aset. Standar ini menekankan pentingnya pengelolaan aset secara holistik untuk mencapai tujuan organisasi.

Manajemen aset juga dapat dilihat dari perspektif bisnis dan keuangan. Buku seperti "Strategic Asset Management: The Quest for Utility Excellence" karya Chris Lloyd menyajikan pandangan strategis dalam mengelola aset untuk meningkatkan kinerja dan memberikan nilai tambah bagi perusahaan. Dengan memahami dan menerapkan prinsip-prinsip manajemen aset, perusahaan dapat mengoptimalkan penggunaan, mengurangi risiko, dan meningkatkan daya saing. Pendekatan ini menjadi semakin penting dalam lingkungan bisnis yang terus berkembang dan kompleks.

2.1.2 Dokumen Manajemen Sistem

Dokumen Manajemen Sistem (DMS) merujuk pada kumpulan dokumen dan informasi yang terkait dengan perancangan, implementasi, operasi, dan pemeliharaan sistem suatu organisasi. DMS memiliki peran krusial dalam memastikan kelancaran dan efektivitas sistem yang digunakan oleh suatu entitas. Ini mencakup dokumentasi mengenai kebijakan, prosedur, panduan teknis, catatan perubahan, serta laporan kinerja sistem.

Dokumen Manajemen Sistem (DMS) adalah suatu sistem komputer atau serangkaian program komputer yang digunakan untuk mengindeks dan menyimpan dokumen elektronik dan gambar dokumen. Penerapan DMS membawa manfaat signifikan dalam meningkatkan efektivitas dan efisiensi proses bisnis atau organisasi. Keuntungan utamanya adalah memungkinkan pengguna untuk dengan cepat menemukan informasi yang diperlukan, menghasilkan proses yang lebih efisien, efektif, dan ekonomis. Sistem manajemen dokumen dirancang untuk menyederhanakan siklus pengelolaan dokumen dalam bisnis atau organisasi, memberikan dukungan otomatisasi dan konsistensi untuk seluruh dokumen kebijakan (Laserfiche, 2007).

2.1.3 Arsip Dokumen Digital

Mengutip dari Srirahayu, 2013:2, Arsip Dokumen Digital atau adalah sebuah sistem atau prosedur untuk mengumpulkan data berupa dokumen yang disimpan menggunakan teknologi komputer dalam bentuk dokumen digital. Tujuan utamanya adalah mempermudah proses melihat, mengelola, menemukan, dan menggunakan kembali dokumen-dokumen tersebut. Menurut National Archives and Record Administration (NASA), arsip elektronik mencakup arsip yang disimpan dan diolah dalam format yang hanya bisa diproses oleh komputer. Electronic records adalah informasi yang tersimpan dalam file dan media digital, yang dibuat, diterima, atau dikelola oleh organisasi atau individu, serta digunakan sebagai bukti aktivitas

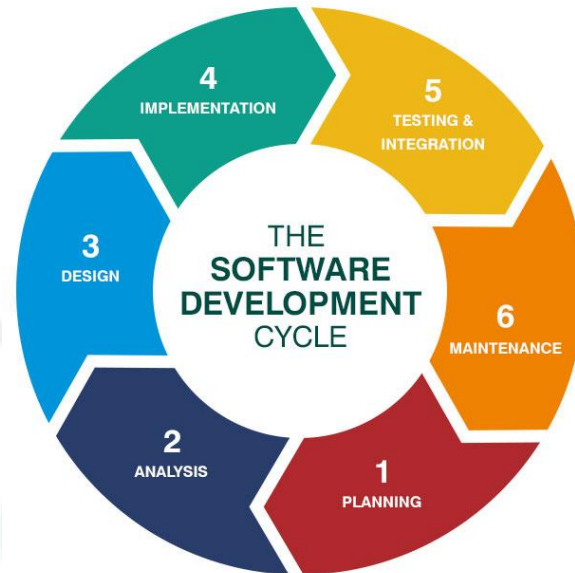
2.1.4 Database

Basis data atau database berfungsi untuk mengatur dan menyimpan data secara lebih efisien dan aman. Sukamto dan Shalahuddin dalam bukunya menyebutkan bahwa database adalah media penyimpanan yang mudah diakses. Sistem Manajemen Basis Data (DBMS) digunakan untuk penyimpanan, pengelolaan, dan penyajian data. Sistem ini memainkan peran penting, termasuk mengatur akses data, menyimpan, memelihara data, mengelola, dan mencegah kebocoran data menggunakan bahasa query yang sesuai. Selain itu, sistem ini memastikan keamanan data. Structured Query Language (SQL) adalah bahasa yang umum digunakan, namun untuk layanan real-time berbasis web yang berhubungan dengan pengumpulan data dan kompleksitas tinggi, diperlukan solusi baru dengan pendekatan NoSQL. Dalam beberapa kasus, keputusan untuk menggunakan atau tidak menggunakan teknologi tertentu harus dipertimbangkan dengan hati-hati. Terkadang, menggabungkan berbagai teknologi mungkin menjadi pilihan terbaik. Pilihan arsitektur database, pengelolaan informasi internal perusahaan harus diatur dan dikembangkan dengan dukungan dari para ahli yang berpengalaman luas (Andreas Meier dan Michael Kaufmann, 2019). Dalam proses perancangan database, normalisasi terjadi. Normalisasi adalah proses yang berkaitan dengan teknik model data relasional dalam desain database (Pujiyanto dan Kadarsih, 2022).

2.1.5 SDLC

Mengutip Binus University Site (2020) mengenai pengertian dan tahapan-tahapan SDLC (System Development Life Cycle), SDLC digunakan untuk membangun sistem informasi agar berfungsi sesuai harapan. SDLC terdiri dari serangkaian tahapan yang saling terkait, yaitu perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Tujuan utama SDLC adalah memastikan pengembangan perangkat lunak yang berkualitas tinggi, efektif, efisien, dan tepat

waktu. Deskripsi dan penjelasan setiap fase SDLC dapat dilihat pada gambar 2.1 SDLC sebagai berikut:



Gambar 2.1 SDLC
Sumber <https://course-net.com/>

1. Perencanaan

Pada fase ini, dilakukan penelusuran terhadap kebutuhan dan permasalahan untuk menciptakan sistem baru atau memperbaiki sistem yang sudah ada. Libatkan pihak terkait seperti programmer, pengguna, stakeholder, dan lainnya untuk mendapatkan berbagai masukan dan pandangan terhadap kelebihan dan kekurangan sistem yang akan dibuat atau diperbaharui.

2. Analisis

Dalam tahap ini, programmer akan merencanakan persyaratan pengembangan sistem, mempertimbangkan risiko yang mungkin terjadi selama proses, termasuk waktu yang diperlukan dan menentukan biaya yang dikeluarkan. Evaluasi strategi pengembangan juga dilakukan untuk menentukan waktu yang optimal.

3. Rancangan / Design

Fase ini melibatkan pembuatan desain sistem berdasarkan kebutuhan klien. Hasil desain diberikan kepada klien untuk pemeriksaan dan umpan balik, memungkinkan klien melihat perkiraan hasil jadi ketika sistem tersebut dibangun.

4. Implementasi

Tahap implementasi terdiri dari beberapa langkah, pertama dengan membangun sistem menggunakan bahasa pemrograman dan menguji sistem untuk mencegah kesalahan. Kemudian, pelatihan pengguna dilakukan dengan berbagai cara, seperti pemotongan langsung, implementasi secara parallel, atau bertahap. Terakhir, perencanaan dukungan sistem baru ditetapkan untuk pengguna setelah implementasi, dan panduan pelaksanaan sistem baru diberikan.

5. Pengujian

Pada tahap ini, sistem yang telah dibangun diuji untuk mengantisipasi kesalahan sebelum digunakan oleh masyarakat. Sistem disesuaikan dengan permintaan klien.

6. Melakukan Pemeliharaan (Maintenance)

Pemeliharaan melibatkan perbaikan kesalahan yang tidak terdeteksi sebelumnya, penyesuaian sistem berdasarkan kebutuhan masyarakat, dan peningkatan kualitas aplikasi.

2.1.6 UML

UML adalah suatu bahasa yang digunakan untuk visualisasi dan dokumentasi hasil pemahaman informasi, memungkinkan proses pembuatan sistem dengan menyajikan gambaran dalam bentuk model dan deskripsi. UML juga didefinisikan sebagai bahasa permodelan yang

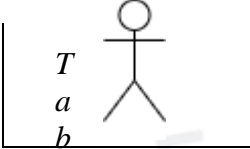
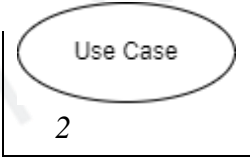
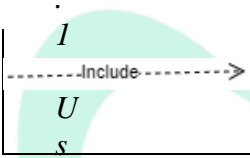
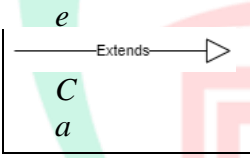
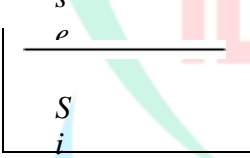
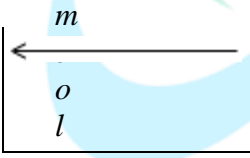

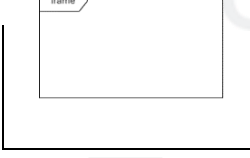
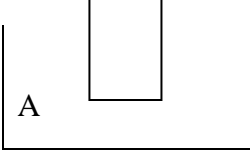
fleksibel, memungkinkan analisis dan penyediaan gambaran keseluruhan dengan notasi yang konsisten di berbagai bidang (Dennis, 2021)

Mengutip dari Nugroho (2014), Unified Modeling Language merupakan bahasa pemodelan yang digunakan untuk sistem atau perangkat lunak dengan konsep berorientasi objek. Penggunaan pemodelan bertujuan menyederhanakan permasalahan yang kompleks sehingga dapat dipelajari dan dipahami dengan lebih mudah. Maka, dapat disimpulkan bahwa UML merupakan alat yang menggunakan grafik atau gambar untuk menggambarkan, menspesifikasikan, dan mendokumentasikan sistem pengembangan perangkat lunak berbasis Object Oriented. Macam UML terdiri dari berikut :

1. Use Case Diagram

Mengutip dari Sukamto dan Shalahuddin, 2014, Diagram use case adalah jenis diagram dalam UML yang digunakan untuk memodelkan fungsionalitas suatu sistem dari perspektif pengguna atau aktor yang berinteraksi dengan sistem. Diagram ini memberikan gambaran umum untuk memodelkan perilaku sistem merupakan pendekatan yang membantu mengidentifikasi persyaratan sistem. Use case dan aktor dalam suatu diagram use case mencerminkan aktivitas yang dilakukan oleh sistem dan cara aktor menggunakannya, tetapi tidak menjelaskan bagaimana sistem bekerja secara internal. Penjelasan mengenai simbol use case beserta deskripsinya pada Tabel 2.1 Use Case Diagram.:



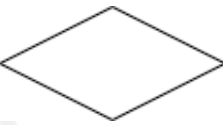
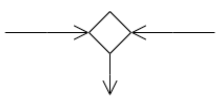



SIMBOL USE CASE DIAGRAM

SIMBOL	NAMA	KETERANGAN
	Actor	Entitas yang berinteraksi dengan sistem. Bisa berupa pengguna manusia atau sistem lain yang berinteraksi dengan sistem yang sedang dimodelkan.
	Use Case	Fungsionalitas atau layanan yang disediakan oleh sistem untuk aktor.
	Include	Menunjukkan bahwa use case target selalu menyertakan perilaku dari use case sumber.
	Extend	Menunjukkan bahwa use case target dapat memperluas perilaku dari use case sumber dalam kondisi tertentu.
	Association	Hubungan antara aktor dan use case. Menunjukkan bahwa aktor berinteraksi dengan use case tersebut.
	Generalization	Hubungan hierarki di mana aktor atau use case anak mewarisi sifat dari aktor atau use case induk.
	Note	Sebagai informasi tambahan yang membantu menjelaskan elemen-elemen dalam diagram.
	Frame	Untuk mengelompokkan elemen-elemen yang berhubungan atau untuk memberikan konteks tambahan.
	System	Mendefinisikan paket yang menampilkan sistem secara terbatas

2. Activity Diagram

Mengutip Fowler, M., & Scott, K. (1997), Activity Diagram merupakan salah satu diagram dalam Unified Modeling Language (UML) yang berfungsi untuk memvisualisasikan alur kerja (workflow) atau aktivitas pada suatu sistem. Activity diagram ini membantu menggambarkan urutan proses atau aktivitas yang dilakukan oleh suatu objek atau sistem untuk mencapai tujuan tertentu.

SIMBOL ACTIVITY DIAGRAM

SIMBOL	NAMA	KETERANGAN
	Initial Node / Start	Menandai titik awal dari sebuah aktivitas. Hanya ada satu initial node dalam setiap activity diagram.
	Activity	Representasi dari sebuah tindakan atau aktivitas. Setiap kotak menunjukkan satu langkah dalam proses.
	Decision Node	Menandai titik di mana sebuah keputusan harus dibuat. Berupa bentuk belah ketupat dengan panah keluar yang menunjukkan jalur yang berbeda berdasarkan kondisi.
	Merge Node	Menggabungkan beberapa aliran masuk menjadi satu aliran keluar. Juga berupa bentuk belah ketupat tetapi dengan beberapa panah masuk dan satu panah keluar.
	Join Node	Menggabungkan beberapa aliran paralel menjadi satu aliran. Digambarkan sebagai garis horizontal atau vertikal dengan beberapa panah masuk dan satu panah keluar.
	Final Node	Menandai titik akhir dari sebuah aktivitas. Digambarkan sebagai lingkaran dengan lingkaran hitam di dalamnya.
	Swimlane	Memisahkan aktivitas berdasarkan siapa yang melakukannya, seperti aktor atau sistem. Membantu mengorganisir diagram dengan jelas.

Tabel 2.2 Activity Diagram Simbol


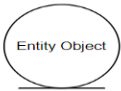
Elemen-elemen utama dalam Diagram Aktivitas:

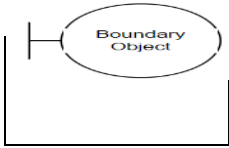
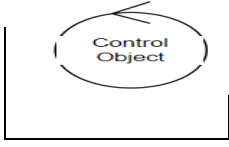
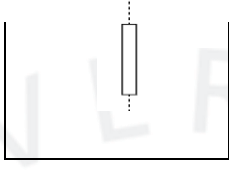
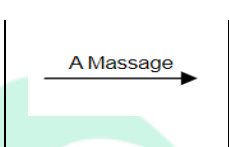
- **Aktivitas (Activity):** Mewakili suatu tindakan atau langkah dalam proses. Contoh aktivitas termasuk operasi, tugas, atau aksi yang harus dilakukan.
- **Garis Aliran (Flow Lines):** Menunjukkan urutan aliran antara aktivitas. Garis aliran menggambarkan bagaimana suatu aktivitas diikuti oleh aktivitas lainnya.
- **Keputusan (Decision):** Mewakili titik dalam proses di mana keputusan harus diambil. Berguna untuk memodelkan kondisi logika atau percabangan dalam aliran kerja.
- **Gabungan (Merge):** Mewakili titik di mana aliran kerja yang terbagi harus digabungkan kembali. Merupakan kebalikan dari keputusan.

3. Sequence Diagram

Sequence diagram digunakan sebagai menggambarkan interaksi antara objek pada suatu sistem atau proses. Diagram urutan ini mengilustrasikan urutan pesan atau panggilan yang terjadi antara objek-objek selama eksekusi suatu fungsi atau proses tertentu. Dengan kata lain, sequence diagram menunjukkan bagaimana objek-objek berinteraksi satu sama lain dalam suatu skenario tertentu.

SIMBOL SEQUENCE DIAGRAM

Simbol	Nama	Keterangan
 <p>Actor</p>	Actor	Menggambarkan orang yang berinteraksi dengan sistem
 <p>Entity Object</p>	Entity Class	Menggambarkan hubungan aktivitas yang akan dilakukan

	Boundary Class	Sebuah penggambaran dari form
	Control Class	Penghubung antara boundary dengan tabel
	A Focus Control & A Life Line	Menggambarkan tempat mulai dan berakhirnya sebuah message
	A Message	Pengiriman pesan

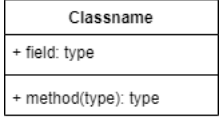
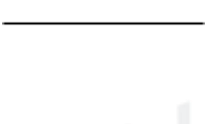


Tabel 2.3 Simbol Sequence Diagram

4. Class Diagram

Class Diagram salah satu jenis diagram dalam UML yang digunakan untuk memodelkan struktur statis dari suatu sistem berorientasi objek. Diagram ini memberikan representasi visual terhadap kelas-kelas, atribut, dan hubungan antar kelas.

Sukanto dan Shalahudin (2013, halaman 141) menciptakan diagram kelas sebagai representasi visual dari struktur sistem, merinci definisi kelas-kelas yang diperlukan dalam pembangunan sistem ini. Diagram kelas berfungsi sebagai panduan bagi pengembang untuk menyusun kelas-kelas sesuai dengan rencana yang terdokumentasi dalam desain perangkat lunak. Diagram ini menampilkan kumpulan kelas, antarmuka, dan kolaborasi yang saling terhubung dalam sistem

Elemen-elemen utama dalam Class Diagram dapat dilihat pada gambar 2.4 *Class Diagram Elemen* dibawah ini:

SIMBOL CLASS DIAGRAM		
SIMBOL	NAMA	KETERANGAN
	Class	Mewakili sebuah kelas dalam sistem dengan nama kelas di bagian atas, diikuti oleh daftar atribut dan metode di bawahnya.
	Association	Menunjukkan hubungan antara dua kelas, seperti bagaimana satu kelas menggunakan atau berhubungan dengan kelas lain.
	Dependency	Menunjukkan kelas yang menggunakan layanan dari kelas lainnya.
	Generalization	Menunjukkan hubungan hierarkis antara kelas induk (superclass) dan kelas turunan (subclass).

Tabel 2.4 Simbol Class Diagram










2.1.7 Flowchart

Flowchart adalah suatu diagram atau representasi grafis yang digunakan untuk menggambarkan langkah-langkah atau alur kerja suatu proses atau sistem.

Menurut Sternecker (Uus Rusmawan, 2019) menyatakan bahwa flowchart atau diagram alir adalah bagan yang menggunakan simbol-simbol grafis untuk menggambarkan aliran algoritma atau proses. Alur tersebut tersusun dari langkah-langkah yang diwakilkan oleh kotak-kotak, dan urutannya ditunjukkan dengan menghubungkan kotak-kotak tersebut menggunakan tanda panah. Diagram ini bermanfaat untuk menyajikan solusi langkah demi langkah untuk menyelesaikan permasalahan dalam suatu proses atau algoritma. Proses atau algoritma tersebut.

Tujuan utama dari flowchart adalah memberikan gambaran visual yang jelas dan mudah dimengerti mengenai bagaimana suatu proses berlangsung, serta menunjukkan urutan langkah-langkah yang harus

diikuti. Flowchart sering digunakan dalam berbagai bidang seperti pemrograman komputer, manajemen operasional, analisis bisnis, dan proses-proses lainnya yang melibatkan langkah-langkah terstruktur.

SIMBOL	NAMA	FUNGSI
	Terminator	Permulaan atau akhiran program
	Input / Output data	Proses input / output data, parameter, dan informasi
	Decision	Pemilihan proses berdasarkan kondisi yang ada
	Process	Pengolahan data yang dilakukan oleh komputer
	On Page Connector	Simbol keluar masuk atau penyambungan proses pada lembar / halaman yang sama
	Off Page Connector	Simbol keluar masuk atau penyambungan proses pada lembar / halaman yang berbeda
	Predefined Process	Permulaan sub program atau proses menjalankan sub program
	Manual Operation	Pengolahan data yang tidak dilakukan oleh komputer
	Dokumen	Menyatakan input yang berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas

Tabel 2.5 Flowchart Simbol

2.1.8 RAD

Mengutip Martin Fowler (2014) Rapid Application Development (RAD) adalah metodologi pengembangan perangkat lunak yang mengutamakan kecepatan dan fleksibilitas. RAD menitikberatkan pada pembuatan prototipe dan iterasi untuk menghasilkan perangkat lunak yang dapat segera digunakan.

RAD, atau prototipe cepat, merupakan metodologi pengembangan perangkat lunak yang berfokus pada siklus pengembangan yang singkat dan berulang. Model ini menggunakan prototipe sebagai model kerja awal untuk mengumpulkan umpan balik pengguna dan meningkatkan desain sistem secara bertahap. Dengan menggunakan pendekatan RAD memungkinkan tim pengembangan untuk beradaptasi dengan perubahan kebutuhan pengguna dengan cepat

dan efisien. RAD cocok untuk proyek dengan persyaratan yang tidak sepenuhnya terdefinisi di awal, di mana umpan balik pengguna sangat penting untuk keberhasilan proyek. (Koestanto,2019).



Gambar 2.2 Model Pengembangan SDLC (RAD)

Metode RAD melibatkan beberapa fase, dimulai dari perencanaan kebutuhan sistem, melibatkan pengguna dalam perancangan sistem, dan proses pembangunan sistem, yang dilakukan secara berulang hingga mencapai kesepakatan bersama. Tahap terakhir adalah implementasi. RAD mengadopsi beberapa model pemodelan, di antaranya Business Modeling, Data Modeling, Process Modeling, Application Generation, Testing, dan Turnover. Keterlibatan aktif pengguna dalam tahap analisis dan desain RAD sangatlah krusial untuk memastikan sistem yang dihasilkan selaras dengan kebutuhan mereka. Hal ini diharapkan dapat mendongkrak tingkat kepuasan pengguna terhadap sistem secara keseluruhan.

RAD merupakan metodologi pengembangan sistem yang berfokus pada empat elemen kunci: Manajemen, Manusia, Metodologi, dan Peralatan. Pendekatan ini menggabungkan berbagai metode dan teknik terstruktur, termasuk prototyping, untuk memahami kebutuhan pengguna dan merancang sistem informasi yang efektif. Proses pengembangan RAD umumnya terdiri dari beberapa langkah yang terdefinisi dengan baik, yaitu:

1. Evaluasi apakah proyek pengembangan sistem memenuhi kriteria yang ditetapkan.
2. Analisis aktivitas bisnis perusahaan untuk menentukan area

- bisnis dan fungsi yang menjadi prioritas.
3. Pembuatan model fungsi-fungsi yang menjadi prioritas.
 4. Pemilihan prototipe yang akan direview.
 5. Implementasi Sistem Informasi.

2.1.9 Pengujian Sistem Black Box

Black box testing merupakan metode pengujian penting dalam pengembangan sistem untuk memverifikasi fungsionalitasnya tanpa perlu memahami detail internal sistem. Berbeda dengan metode lain yang meneliti kode internal, black box testing berfokus pada "apa" yang dilakukan sistem, bukan "bagaimana" sistem tersebut bekerja. Pengujian hanya perlu mengetahui masukan yang diberikan dan keluaran yang diharapkan, tanpa perlu pusing memikirkan arsitektur rumit di balik layar. Fleksibilitasnya memungkinkan black box testing diintegrasikan ke berbagai tahap siklus hidup pengembangan, mulai dari pengujian unit terkecil hingga pengujian keseluruhan sistem. Analogi mesin pencari dengan jelas menunjukkan prinsip black box testing. Pengujian, seperti pengguna mesin pencari, tidak perlu memahami detail algoritma pencarian, melainkan hanya fokus pada masukan (kata kunci) dan keluaran (hasil pencarian). (M.e Khan,2011) dalam (Leksanti, 2020)

2.1.10 Pengumpulan Data

Pengumpulan data dokumen manajemen sistem bertujuan untuk mendapatkan informasi yang akurat dan relevan guna mendukung pengambilan keputusan, pemecahan masalah, dan peningkatan kinerja organisasi. Metode Pengumpulan data ini terdiri dari :

1) Survei dan Kuesioner

Metode ini melibatkan penyebaran kuesioner kepada pengguna sistem atau pihak terkait untuk mengumpulkan data tentang penggunaan dan kinerja sistem. Kuesioner

dirancang untuk mendapatkan tanggapan yang dapat diukur dan dianalisis secara statistik.(Fowler, F. J. (2013). *Survey Research Methods* (5th ed.). Sage Publications.)

2) **Wawancara**

Wawancara mendalam dengan pengguna sistem, manajer, dan staf IT dapat memberikan wawasan yang lebih dalam tentang bagaimana sistem berfungsi dan area yang memerlukan perbaikan.(Kvale, S., & Brinkmann, S. (2009). *InterViews: Learning the Craft of Qualitative Research Interviewing* (2nd ed.). Sage Publications.)

3) **Observasi**

Observasi langsung terhadap penggunaan sistem di tempat kerja dapat membantu peneliti memahami konteks penggunaan dan mengidentifikasi masalah yang mungkin tidak terdeteksi melalui metode lain. (Adler, P. A., & Adler, P. (1994). *Observational Techniques*. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of Qualitative Research*. Sage Publications)

4) **Analisis Dokumen**

Mengkaji dokumen yang ada, seperti laporan kinerja, kebijakan, dan prosedur, untuk memahami bagaimana sistem dikelola dan diimplementasikan. (Bowen, G. A. (2009). *Document Analysis as a Qualitative Research Method*. *Qualitative Research Journal*, 9(2), 27-40.)

2.2. Tinjauan Studi

Tinjauan studi Tinjauan studi memiliki peran penting bagi peneliti sebagai panduan dan pendukung dalam penyusunan tugas akhir. Di bawah ini adalah daftar referensi yang digunakan dalam penelitian ini dan memiliki keterkaitan

dengan topik penelitian, sebagai berikut:

1. Penelitian oleh (Setyawati, 2023) dengan judul “Desain Sistem Informasi Penanganan Arsip Berbasis Website di PT Sinar Nusrapress Utama” bertujuan untuk mempermudah, mempercepat, dan mengatasi masalah pengarsipan surat masuk dan keluar di PT Sinar Nusrapress Utama, sebuah perusahaan media di Denpasar, Bali. Solusi yang diusulkan adalah merancang sistem penanganan arsip berbasis website yang menggunakan sistem informasi untuk meningkatkan efektivitas dan efisiensi pengarsipan. Metode yang digunakan adalah System Development Life Cycle (SDLC), yang mencakup identifikasi masalah, analisis kebutuhan pengguna, analisis sistem, dan desain sistem. Alat bantu yang digunakan meliputi concept map, entity relationship diagram, use case diagram, flowchart, dan GUI, dengan teknik pengumpulan data melalui observasi, wawancara, studi pustaka, dan dokumentasi. Hasilnya adalah desain sistem informasi arsip berbasis website yang lebih efektif dan efisien.
2. Penelitian oleh Dimas Rifqi Ekaryanto dan Siti Ramadhani (2021) “RANCANG BANGUN SISTEM INFORMASI DIGITALISASI MANAJEMEN DOKUMEN PADA RUMAH SAKIT AWAL BROS PEKANBARU” membahas implementasi Sistem Informasi Digitalisasi Manajemen Dokumen di Rumah Sakit Awal Bros Pekanbaru. Pentingnya manajemen dokumen di rumah sakit ditekankan, dan solusi berupa sistem berbasis web dirancang menggunakan framework CodeIgniter dan MySQLI. Pengujian sistem dilakukan dengan metode black box, dan hasilnya mencapai 100% keberhasilan fungsi sistem. User Acceptance Test (UAT) memberikan skor "Sangat Setuju" sebesar 86,4%, menunjukkan tingkat penerimaan yang tinggi dari pengguna. Sistem ini memberikan manfaat signifikan, memungkinkan akses digital, mengurangi penumpukan fisik dokumen, dan mempermudah pengarsipan. Kesimpulan menegaskan bahwa sistem berhasil memenuhi kebutuhan dan mempermudah pengelolaan dokumen di rumah sakit.

3. Studi oleh Nugraha (2021) berjudul "Aplikasi Pengarsipan Dokumen dan Data Inventaris pada Kantor Kelurahan Pekapuran Raya Kota Banjarmasin yang Berbasis Web" menyoroti kebutuhan Kantor Kelurahan Pekapuran Raya Kota Banjarmasin akan sistem pengarsipan dokumen dan data inventaris yang lebih terorganisir, mengingat saat ini proses tersebut masih dilakukan secara manual menggunakan Microsoft Office. Sistem baru ini dirancang menggunakan Sublime Text 3 dengan bahasa pemrograman HTML, PHP, dan database MySQL. Sistem ini diharapkan dapat mempermudah dan mempercepat proses pengarsipan dan pendataan inventaris. Disarankan juga untuk melakukan monitoring disposisi surat masuk dan pemeliharaan berkala terhadap aplikasi agar sistem tetap berjalan dengan baik.
4. 4. Muhammad Iqbal Zaunedi, Muharman Lubis, Ahmad Musnansyah "Pengembangan DMS (Document Management System) Berbasis Web di Fakultas Rekayasa Industri Universitas Telkom" membahas pengembangan sistem manajemen dokumen berbasis web di Fakultas Rekayasa Industri Universitas Telkom. Aplikasi yang dinamakan EzDesk ini dirancang untuk membantu reviewer dalam meninjau karya tulis atau paper pada konferensi. EzDesk melibatkan empat aktor utama: Penulis (Author), Reviewer, Ketua (Chair), dan Admin. Aplikasi ini menyediakan fitur-fitur seperti tampilan (view) dan komentar (comment) yang mempermudah proses peninjauan. Kesimpulan dari penelitian ini mencakup penambahan fitur untuk membantu reviewer, seperti tampilan dan komentar, dengan tujuan meningkatkan efisiensi proses review.
5. Jurnal oleh Nur Khotimah, "PENGEMBANGAN SISTEM PENCATATAN ARSIP DINAMIS BERBASIS WEBSITE DI KANTOR KECAMATAN GUNUNGPATI SEMARANG" (2020) menyoroti pentingnya manajemen arsip dalam era teknologi. Arsip, sebagai sumber informasi, membutuhkan manajemen efektif, terutama dengan jumlah yang semakin meningkat. Undang-Undang kearsipan menjadi landasan, dengan penekanan pada

prinsip, kaidah, dan standar kearsipan nasional yang andal. Kendala dalam pengelolaan arsip di Kantor Kecamatan Gunungpati Semarang termasuk penyimpanan manual dan kurangnya pengetahuan. Konsep "office paperless" dan e-Government diusulkan sebagai solusi. Penelitian menyoroti kegiatan sosialisasi dan workshop sebagai persiapan sumber daya manusia menghadapi perkembangan teknologi. Kesimpulannya, skripsi ini mengembangkan Sistem Pencatatan Arsip Dinamis Berbasis Website di Kantor Kecamatan Gunungpati Semarang untuk meningkatkan efisiensi dan efektivitas.

