

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 Sistem

Menurut perspektif Bambang Hartono (2017), suatu sistem dapat didefinisikan sebagai kumpulan elemen atau bagian yang terorganisir dan saling terkait berdasarkan fungsi-fungsinya. Pendekatannya menekankan signifikansi memberikan perhatian pada setiap komponen yang membentuk sistem, karena setiap bagian tersebut memiliki peran yang vital dalam mengelola suatu organisasi.

Jogiyanto (2017) mengungkapkan:

Bahwa suatu sistem mempunyai karakteristik yaitu

1. **Komponen Sistem**

Sebuah sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerja sama untuk membentuk satu keseluruhan. Elemen-elemen atau komponen dalam sistem dapat berwujud sebagai subsistem atau bagian-bagian yang tidak terpisahkan dari sistem itu sendiri. Setiap subsistem memiliki sifat-sifat khusus sebagai sistem untuk melaksanakan fungsi tertentu dan mengatur sistem secara menyeluruh.

2. **Batas Sistem**

Batasan sistem mengacu pada area yang menentukan batas antara sistem dan sistem lain atau lingkungan eksternal. Adanya batas-batas sistem memungkinkan kita memandang sistem sebagai suatu kesatuan yang terpisah.

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

Lingkungan eksternal suatu sistem merujuk pada semua elemen di luar batas sistem yang memiliki dampak terhadap fungsi operasional sistem tersebut. Lingkungan luar sistem

dapat memberikan pengaruh positif (memerlukan perhatian dan merupakan sumber energi bagi sistem) atau dapat memberikan pengaruh negatif (memerlukan kewaspadaan dan pengendalian).

4. Penghubung Sistem (*Interface*)

Penghubung merupakan alat penghubung antara satu subsistem dan subsistem lainnya. Dengan adanya penghubung ini, sumber daya dapat mengalir dari satu subsistem ke subsistem lainnya. Output dari satu subsistem menjadi input bagi subsistem lain melalui penghubung, menciptakan interaksi antara subsistem-subsistem tersebut sehingga membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Mengacu pada energi yang dimasukkan ke dalam sistem. Jenis masukan dapat terdiri dari masukan perawatan (input perawatan) dan masukan sinyal (input sinyal). Input perawatan adalah energi yang diperlukan untuk menjaga agar sistem dapat beroperasi. Sementara itu, input sinyal adalah energi yang diolah untuk menghasilkan keluaran.

6. Keluaran Sistem (*Output*)

Output adalah hasil dari energi yang diolah dan diuraikan menjadi keluaran yang bermanfaat serta sisa pembuangan. Keluaran dapat berperan sebagai input untuk subsistem lain atau untuk super sistem.

7. Pengolah Sistem

Sebuah sistem mengandung elemen pemroses yang berfungsi untuk mengubah input menjadi output.

8. Sasaran Sistem

Setiap sistem memiliki sasaran atau target yang harus dicapai. Tanpa adanya tujuan, operasi sistem kehilangan makna.

Sukses atau keberhasilan suatu sistem dinilai berdasarkan sejauh mana sistem tersebut mencapai sasaran atau targetnya.

2.1.2 Informasi

Menurut Goh dan Yap (2017), informasi adalah "sumber daya yang diperoleh dari pemrosesan data berkualitas, yang dapat digunakan untuk membuat keputusan atau menghasilkan pengetahuan yang berguna".

2.1.3 Sistem Informasi

Sistem informasi dapat dijelaskan sebagai sekumpulan komponen terintegrasi yang melibatkan perangkat keras, perangkat lunak, basis data, jaringan, dan partisipasi manusia. Sistem ini dirancang untuk melakukan pengumpulan, pemrosesan, penyimpanan, dan distribusi informasi, dengan tujuan memberikan dukungan bagi pengambilan keputusan dan manajemen dalam suatu organisasi."

2.1.3 Website

Website merupakan sebuah platform yang berisikan beberapa halaman web yang memiliki isi atau topik pembahasan yang saling berhubungan satu sama lain. Penyimpanan data dari website itu sendiri biasanya berada Di sebuah server web yang dapat dijangkau melalui jaringan internet, demikian seperti yang disampaikan oleh Yeni Susilowati pada tahun 2019. Dengan adanya situs web, akses terhadap berbagai informasi menjadi lebih mudah. Cukup dengan menggunakan perangkat seperti gadget, smartphone, atau komputer, situs web dapat diakses kapan saja dan di mana saja, tentunya dengan memanfaatkan koneksi internet sebagai salah satu elemen utama dalam mengaksesnya.

2.1.5 Database

Database atau basis data merujuk pada himpunan data yang tersimpan secara terstruktur dalam suatu sistem komputer dan dapat diakses serta dikelola oleh perangkat lunak atau aplikasi tertentu. Dalam perspektif

Connolly dan Begg (2018), database merupakan koleksi data yang terstruktur, terorganisir, dan terintegrasi dengan baik, memungkinkan akses dan manajemen yang mudah. Database juga dapat dijelaskan sebagai sistem yang mampu menyimpan, memanipulasi, dan mengelola data secara efisien.

2.1.6 Pengertian Waterfall

Model Waterfall merupakan suatu pendekatan pengembangan sistem yang terstruktur dan berurutan dalam membangun perangkat lunak. Proses ini melibatkan beberapa fase, seperti analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan, sebagaimana dijelaskan oleh Pressman (2015). Dalam metode pengembangan Waterfall, empat tahap pokok harus dilalui, yakni Analisis Kebutuhan, Desain, Implementasi, Pengujian, dan Pemeliharaan.

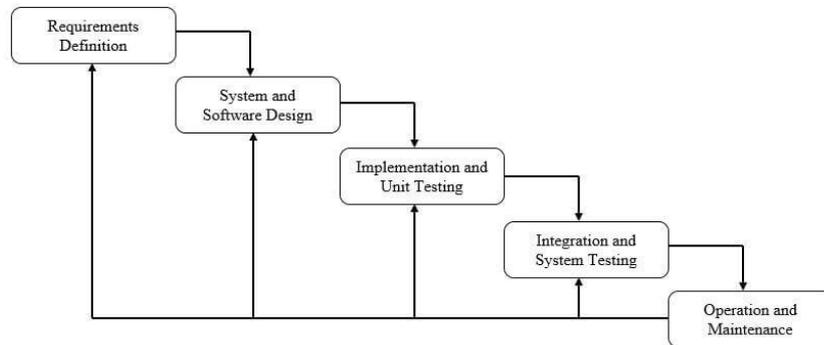
2.1.7 Pengertian SDLC (System Development Life-Cycle)

Siklus Hidup Pengembangan Perangkat Lunak (SDLC) merupakan suatu metode atau pendekatan dalam pengembangan perangkat lunak yang melibatkan serangkaian tahap atau fase yang terstruktur dan saling terkait. Menurut definisi dari Pressman (2018).

SDLC adalah suatu kerangka kerja yang terstruktur untuk pengembangan perangkat lunak yang efektif dan efisien. SDLC mencakup serangkaian tahap yang saling terkait, dimulai dari perencanaan, analisis, desain, implementasi, hingga pemeliharaan. Dalam pelaksanaan prosesnya.

SDLC memiliki berbagai model, dan salah satu model yang umum dan tertua adalah model Waterfall. Menurut Kusnadi (2019), model Waterfall dalam SDLC adalah metode pengembangan perangkat lunak yang linier dan berurutan dimana setiap fase dijalankan satu demi satu dan tidak ada cara untuk kembali ke fase sebelumnya. Model Waterfall terdiri dari beberapa tahapan yang meliputi perencanaan,

analisis, desain, implementasi, pengujian dan pemeliharaan. Terdapat gambar model air terjun. pada Gambar 2.1 di bawah ini:



Gambar 2.1 Metode Waterfall

Menurut Ian Sommerville (2020), tahapan dalam model Waterfall terdiri dari enam fase:

1. Analisis Kebutuhan (*Requirements Analysis*)
Pada fase ini, tim pengembang melakukan pengumpulan dan analisis kebutuhan sistem. Hasil dari tahap ini adalah dokumen analisis kebutuhan yang berisi kebutuhan fungsional dan non-fungsional dari sistem.
2. Desain (*Design*)
Pada tahap ini, tim pengembang merancang arsitektur sistem dan mengembangkan spesifikasi rinci dari sistem. Hasil dari tahap ini adalah dokumen desain sistem.
3. Implementasi (*Implementation*)
Pada tahap ini, seluruh komponen sistem diintegrasikan menjadi satu kesatuan dan diuji secara menyeluruh. Hasil dari tahap ini adalah sistem yang siap untuk diuji.
4. Integrasi (*Integration*)
Pada fase ini, sistem diuji dengan tujuan memastikan bahwa kinerjanya sesuai dengan spesifikasi dan memenuhi kebutuhan pengguna. Proses pengujian mencakup uji fungsional, uji kinerja, dan uji integrasi.
5. Operasi dan Pemeliharaan (*Operation and Maintenance*)

Pada tahap ini, sistem dirilis dan digunakan oleh pengguna. Tim pengembang akan melakukan pemeliharaan dan perbaikan pada sistem jika terjadi masalah.

2.1.8 Unified Modeling Language (UML)

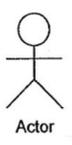
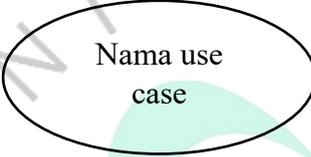
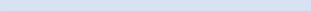
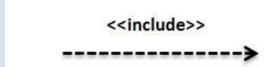
Unified Modeling Language (UML) adalah suatu bahasa pemodelan visual yang dipakai untuk perancangan, pengembangan, dan dokumentasi perangkat lunak yang berbasis objek. Pressman (2018) mengartikan UML sebagai bahasa pemodelan visual yang memvisualisasikan struktur, perilaku, dan interaksi sistem yang berbasis objek. UML berperan sebagai bahasa standar yang dapat digunakan oleh semua pihak yang terlibat dalam pengembangan perangkat lunak, termasuk analis, desainer, dan pemrogram. UML mengadopsi konsep dari Pemrograman Berorientasi Objek (OOP) seperti objek, kelas, pewarisan, dan polimorfisme untuk merepresentasikan sistem secara visual menggunakan diagram dan model. Ada berbagai jenis diagram UML untuk memodelkan berbagai aspek sistem berorientasi objek, seperti diagram kelas, diagram objek, diagram fungsi, diagram urutan.

2.1.9 Use Case Diagram

Diagram Use Case merupakan jenis diagram UML (Unified Modeling Language) yang dimanfaatkan untuk menggambarkan fungsionalitas atau fitur-fitur utama (use case) suatu sistem dari perspektif pengguna atau aktor. Use case diagram berfungsi untuk menyajikan keterkaitan antara aktor dengan use case terkait, serta interaksi antara use case satu dengan yang lain dalam sistem. Diagram ini membantu dalam pemahaman terhadap persyaratan sistem dan juga berguna untuk mengidentifikasi serta mengorganisir fitur-fitur utama dalam sistem yang sedang dikembangkan, seperti dijelaskan oleh Fowler (2018). Tabel 2.1 berisi simbol-simbol yang digunakan dalam use case diagram.

1. Use Case Diagram

Tabel 2.1 Use Case Diagram

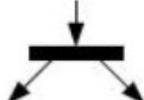
No.	Simbol	Nama	Keterangan
1		Actor	Seseorang, suatu proses, atau sistem lain yang mampu berinteraksi dengan lingkungan di luar dari sistem informasi yang sedang dibangun.
2		Use Case	Sebuah fungsi berperan sebagai elemen yang saling berinteraksi satu sama lain atau dengan aktor-aktor lainnya.
3		Association	Garis penghubung digunakan untuk melakukan komunikasi antara use case dan aktor.
4		Extend	Hubungan tambahan pada use case yang dapat berdiri sendiri tanpa ketergantungan pada use case lain.
5		Include	Use case tambahan yang membutuhkan ketergantungan pada use case lain
6		Generalization	relasi antara 2 (dua) buah use case secara umum, salah satunya menjadi fungsi lebih umum dari lainnya.

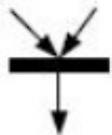
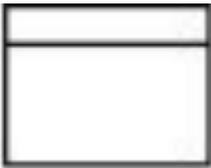
2. Activity Diagram

Activity diagram adalah salah satu jenis diagram UML (*Unified Modeling Language*) yang digunakan untuk merepresentasikan urutan kegiatan atau aktivitas dalam sebuah proses bisnis atau sistem

perangkat lunak. Diagram ini dapat digunakan untuk memodelkan alur kerja suatu sistem dengan jelas, termasuk alur percabangan (*branching*) dan pengulangan (*looping*). *Activity* diagram terdiri dari berbagai elemen, seperti aktivitas (*activity*), keputusan (*decision*), garis hubung (*flow*), dan titik awal/titik akhir (*start/end point*). Aktivitas dalam diagram ini merepresentasikan sebuah tindakan atau aktivitas yang terjadi dalam suatu sistem, sedangkan keputusan merepresentasikan percabangan dalam aktivitas atau pilihan-pilihan yang harus dibuat oleh pengguna sistem. Dibawah ini merupakan **tabel 2.2 Activity Diagram**.

Tabel 2.2 Activity Diagram

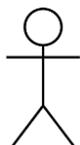
Simbol	Nama	Keterangan
	Initial Node	Simbol yang digunakan untuk memulai <i>Activity Diagram</i> .
	Activity	Menunjukkan lambang yang mencerminkan kegiatan atau tugas yang sedang dilakukan dalam suatu alur kerja.
	Decision	Lambang yang digunakan ketika terdapat opsi atau pilihan kegiatan.
	Line Connector	Dipakai untuk mengaitkan antara satu lambang dengan lambang lainnya
	Final Node	Lambang yang digunakan untuk menyelesaikan sebuah diagram aktivitas.
	Fork	digunakan untuk memisahkan alur menjadi beberapa alur secara simultan.

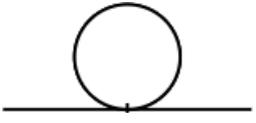
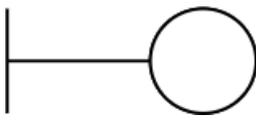
	Join	Join berperan sebagai titik konvergensi untuk proses-proses yang menggabungkan beberapa aliran menjadi satu aliran..
	Swim Lane	Pemisahan alur bisnis berdasarkan kegiatan yang terjadi.

3. Sequence Diagram

UML (Unified Modeling Language) yang berfungsi untuk menggambarkan interaksi atau hubungan di antara objek-objek dalam suatu sistem perangkat lunak atau proses bisnis. Diagram ini memperlihatkan urutan pesan yang dikirim antara objek-objek dalam sistem, serta menyajikan informasi mengenai waktu atau urutan kejadian setiap pesan yang terkirim. *Sequence* diagram terdiri dari objek-objek, pesan, dan waktu atau urutan kejadian yang diwakili oleh garis vertikal yang disebut lifeline. Dibawah ini merupakan **Gambar 2.3** *Sequence* Diagram.

Tabel 2.3 Sequence Diagram

Simbol	Nama	Keterangan
	Actor	Sebuah elemen di luar sistem yang menggambarkan pengguna dan berinteraksi dengan sistem.

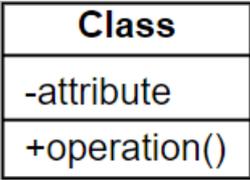
	Entity Class	Adalah bagian dari sistem yang terdiri dari sekumpulan kelas.
	Boundary Class	Interaksi antara <i>actor</i> dengan sistem yang menggambarkan UI (<i>User Interface</i>).
	Control Class	Digunakan untuk mengilustrasikan logika aplikasi yang tidak memiliki tanggung jawab terhadap entitas.
	Message	Ini memvisualisasikan pesan atau interaksi antara objek.

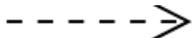
	Recursive	Menggambarkan mengirim suatu pesan yang dikirim untuk objek itu sendiri.
	Lifeline	Menggambarkan garis hidup dari sebuah objek.
	A Focus of Control (Activation)	Untuk menunjukkan periode selama objek melakukan aktifitas.

4. Class Diagram

Diagram kelas merupakan salah satu jenis diagram UML (Unified Modeling Language) yang berfungsi untuk menggambarkan struktur kelas dalam suatu sistem perangkat lunak atau aplikasi. Diagram ini memperlihatkan kelas-kelas yang ada dalam sistem dan relasi di antara kelas-kelas tersebut. Class diagram terdiri dari kelas, atribut, metode, dan hubungan antara kelas-kelas tersebut. Di bawah ini merupakan **Gambar 2.4** Class Diagram

Tabel 2.4 Class Diagram

Simbol	Nama	Keterangan
	Class	<p>Kelas adalah deskripsi struktur sistem yang berisi properti dan fungsi.</p> <p>Atribut adalah properti suatu class</p>

	Association	Hubungan adalah keterkaitan antara dua kelas atau lebih yang mencerminkan bahwa kelas-kelas tersebut berinteraksi
	Generalization	Keterkaitan antara dua atau lebih kelas yang mencerminkan bahwa kelas-kelas tersebut mempunyai atribut atau perilaku serupa dan terstruktur dalam hierarki atau tingkatan yang berbeda dalam suatu sistem perangkat lunak
	Aggregation	Hubungan antara dua kelas atau lebih yang menunjukkan bahwa kelas-kelas ini membentuk unit yang lebih besar dalam perangkat lunak.
	Composition	Suatu hubungan antara dua kelas dimana satu kelas merupakan bagian penting dari kelas lainnya dan tidak dapat ada atau ada secara mandiri tanpa kelas lainnya.
	Dependency	hubungan antara dua kelas di mana perubahan pada kelas yang satu akan mempengaruhi kelas yang lainnya.

2.2 Tinjauan Pustaka

Tinjauan Pustaka dilakukan untuk memberikan penulis pemahaman yang lebih mendalam tentang teori atau metode tertentu, serta sebagai sumber referensi untuk memperoleh teori-teori yang relevan dengan penelitian saat ini. Beberapa referensi yang digunakan sebagai pendukung dalam penelitian ini mencakup:

1. Penelitian yang dilakukan oleh Hery, Nathanael, & Widjaja (2021) bertujuan untuk mengakomodasi sebagian besar informasi yang biasanya disampaikan kepada jemaat melalui pencatatan. Saat ini, gereja juga mendorong praktik lingkungan hijau dengan mengurangi penggunaan kertas dan beralih ke teknologi informasi untuk mendukung aktivitas gereja. Pendekatan waterfall dalam penelitian ini mengacu pada konsep air terjun, memberikan arahan yang jelas dalam pengumpulan informasi hingga pembuatan situs web gereja. Hasil penelitian ini mencakup bantuan kepada jemaat dalam penyediaan informasi digital, Data kehadiran jemaat, daftar petugas dan tanggung jawab mereka, rincian tentang pendeta dan isi khotbah, bersama dengan laporan keuangan sumbangan, merupakan beberapa contoh informasi yang dapat diakses. Selain Selain itu, berbagai informasi yang terkait dengan kegiatan gereja juga dapat diakses dan diunduh melalui halaman web..
2. Penelitian yang dilakukan oleh Sagala, Sadikin, & Irawan (2018) menunjukkan bahwa GKPI Palmerah Jambi masih menggunakan Aplikasi Microsoft Word digunakan untuk mengelola berbagai data di gereja, termasuk data jemaat, pembaptisan, peneguhan sidi, kematian, dan pernikahan. Data mengenai jadwal ibadah, kegiatan gereja, dan acara ibadah lainnya masih diberikan melalui publikasi berbasis kertas dan papan informasi (mading). Proses pencetakan warta ibadah dan laporan memerlukan alokasi dana operasional yang signifikan. Dalam upaya merancang sistem informasi gereja berbasis web, digunakan metode

Object Oriented Analysis and Design (OOAD). Konsep OOAD melibatkan analisis dan desain sistem dengan pendekatan objek, di mana OOA bertujuan untuk memeriksa persyaratan (requirement), sementara OOD digunakan untuk membimbing arsitektur perangkat lunak dengan memanipulasi objek-objek sistem atau subsistem.

3. Penelitian yang dilakukan oleh Sinaga (2018) Membicarakan kepentingan fasilitas dan infrastruktur yang mendukung tugas administrative. Hal ini diperlukan untuk memperkuat proses pelayanan gereja dan meningkatkan kualitas lembaga keagamaan tersebut. Kesalahan dalam pencatatan dan manajemen pelayanan dapat berdampak pada peningkatan biaya operasional dan ketidakstabilan keuangan gereja. Aplikasi ini dikembangkan menggunakan bahasa pemrograman PHP dan data disimpan dalam database MySQL. Dilengkapi dengan berbagai fitur, aplikasi ini bertujuan memberikan kenyamanan kepada pengguna dalam mengelola pelayan gereja dan menyusun laporan data keuangan gereja secara komprehensif. Dengan menggunakan aplikasi ini, pengguna dapat mencatat penerimaan pelayan, mengelola informasi pelayan, dan menyusun laporan data pelayan serta keuangan gereja secara menyeluruh.
4. Penelitian yang dilakukan oleh Patricia Stephanie Gloria Padoma pada tahun 2021 mengenai pengembangan sistem informasi perencanaan program kerja berbasis web. Penelitian ini secara khusus menganalisis studi kasus yang melibatkan Jemaat Gereja Pantekosta Kalvari Indonesia (GKPMI) Getsemani di Sorong. Dari hasil studi kasus ini dapat disimpulkan bahwa paroki memerlukan suatu sistem informasi untuk mengontrol dan mengelola program kerja yang dibuat melalui rapat pengurus paroki. Sistem yang diimplementasikan menggunakan bahasa pemrograman PHP dengan framework CodeIgniter dan database MySQL.
5. Penelitian pada tahun 2023 yang dilakukan oleh Nidia Rosmawant, Maya Angrein dan Muslihuddin bertujuan untuk membangun model pelayanan gereja online yang meningkatkan efektivitas dan efisiensi pelayanan gereja. Pendekatan yang digunakan untuk mengatasi tantangan ini adalah

penelitian dan pengembangan (RandD), suatu proses pengembangan alat pendidikan yang melibatkan banyak penelitian menggunakan metode berbeda pada tahapan berbeda. Tujuan pendekatan ini adalah untuk mengontrol dan meningkatkan kualitas penelitian. Hasil penerapan kuesioner yang disebar menunjukkan seluruh pertanyaan valid dan sebagian responden setuju dengan persentase sangat setuju 50,75%, setuju 39% dan agak setuju 10,25%.

