

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Pada penelitian ini, penulis melakukan kajian terhadap berbagai teori yang akan dijadikan acuan dan dasar dalam penelitian ini, yang bertujuan untuk merancang dan membangun aplikasi pengelolaan proyek dan beban kerja di perusahaan konsultan XYZ, antara lain sebagai berikut:

2.1.1 Rancang Bangun

Menurut Pressman & Maxim (2019), rancang merupakan tahap di mana hasil analisis sistem diterjemahkan ke dalam bahasa pemrograman untuk merinci implementasi komponen-komponen sistem. Sementara itu, bangun merujuk pada aktivitas yang melibatkan penciptaan, perubahan, atau perbaikan sistem yang ada. Dengan demikian, rancang bangun dapat dipahami sebagai proses analisis dan perencanaan yang bertujuan untuk mentransformasikan sistem ke dalam bahasa pemrograman guna menciptakan atau memperbaiki sistem tersebut.

Rancang bangun perangkat lunak melibatkan analisis dan perencanaan yang terstruktur, di mana kerangka kerja SDLC (*System Development Life Cycle*) digunakan untuk mengelola tahapan pengembangan dari analisis hingga pemeliharaan. Pendekatan OOAD (*Object-Oriented Analysis and Design*) berperan penting dalam merancang sistem berbasis objek, yang memudahkan penerjemahan hasil analisis ke dalam model yang siap untuk diimplementasikan dalam kode. Pemilihan metode pengembangan, seperti *Agile* atau *Waterfall*, disesuaikan dengan kebutuhan proyek untuk memastikan setiap fase berjalan dengan lancar dan memenuhi tujuan yang telah ditetapkan.

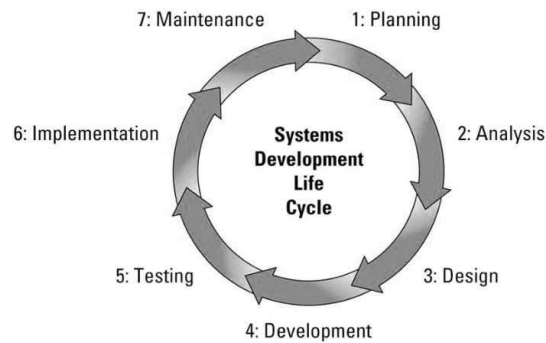
A. *System Development Life Cycle*

Menurut Pressman dan Maxim (2019), *System Development Life Cycle* (SDLC) adalah Proses terstruktur yang mencakup semua fase dalam pengembangan perangkat lunak, mulai dari pengumpulan persyaratan hingga tahap pemeliharaan. Tujuan dari SDLC adalah

memastikan bahwa perangkat lunak yang dikembangkan memiliki kualitas yang baik, mampu memenuhi kebutuhan pengguna, dan diselesaikan tepat waktu serta sesuai anggaran yang telah ditetapkan.

Terdapat beberapa fase dalam *System Development Life Cycle* (SDLC) yang saling terkait, mulai dari perencanaan hingga pemeliharaan perangkat lunak:

- **Perencanaan:** Identifikasi kebutuhan dan tujuan proyek, analisis kelayakan (teknis, ekonomi, operasional), penyusunan jadwal, anggaran, serta pembentukan tim dengan peran yang jelas.
- **Analisis Kebutuhan:** Pengumpulan kebutuhan pengguna melalui wawancara dan survei, penyusunan dokumen SRS, serta pembuatan model kebutuhan, seperti *Use Case* diagram.
- **Desain Sistem:** Perancangan arsitektur perangkat lunak, desain antarmuka pengguna, dan basis data untuk kemudahan penggunaan serta integritas data.
- **Pengembangan (*Development*):** Proses pengembangan sistem berdasarkan desain, pengujian unit untuk verifikasi, serta dokumentasi untuk mendukung pemeliharaan.
- **Pengujian:** Pengujian fungsi, integrasi, dan non-fungsional sistem, termasuk UAT untuk memastikan sistem sesuai harapan pengguna.
- **Penerapan (*Deployment*):** Peluncuran sistem, pelatihan pengguna, migrasi data, dan pemantauan awal untuk memastikan kelancaran implementasi.
- **Pemeliharaan:** Perbaikan bug, pembaruan sistem, pengumpulan umpan balik, dan penyesuaian dengan perkembangan teknologi.



Gambar 2.1 System Development Life Cycle

B. Object Oriented Analysis and Design






Menurut Pressman dan Maxim (2019) Analisis dan Desain Berorientasi Objek (OOAD) merupakan pendekatan dalam pengembangan perangkat lunak yang menggunakan konsep objek untuk merancang sistem yang kompleks. Pendekatan ini memfasilitasi pemodelan dunia nyata, menjadikannya lebih intuitif dan mudah dipahami.

C. Unified Modeling Language

UML umumnya menyediakan model grafis dari sebuah sistem dengan menggunakan berbagai kelompok berdasarkan fungsionalitasnya. Semua diagram atau model UML yang dibuat oleh ahli pengembangan didasarkan pada berbagai perspektif dari sistem data dan informasi (Purwaningtias, 2018). Diagram-diagram yang digunakan dalam UML, antara lain:

1. Use Case Diagram







Use Case Diagram menggambarkan hubungan antara sistem internal dan eksternal, termasuk pengguna akhir. Diagram ini juga secara visual menunjukkan penerapan model tersebut serta berbagai cara pengguna berinteraksi dengan sistem yang dimaksud.

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Menggambarkan himpunan peran yang berinteraksi dengan Use Case.
2.		<i>Dependency</i>	Relasi yang menggambarkan ketergantungan antara Use Case satu dengan Use Case lainnya.
3.		<i>Generalization</i>	Relasi yang menggambarkan turunan (<i>inheritance</i>) dan meng- <i>override</i> sifat dari objek lainnya, baik Aktor maupun Use Case.
4.		<i>Include</i>	Menunjukkan bahwa suatu Use Case seluruhnya merupakan fungsionalitas dari Use Case lainnya.
5.		<i>Extend</i>	Menunjukkan bahwa suatu Use Case merupakan tambahan fungsional dari Use Case lainnya apabila suatu kondisi terpenuhi.

Gambar 2.2 Notasi pada Use Case Diagram

2. Activity Diagram

- *Activity Diagram* digunakan untuk memodelkan aliran data secara berurutan dari berbagai aktivitas, baik itu proses bisnis maupun use case tertentu.




NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Activity</i>	Memperlihatkan bagaimana setiap kelas antarmuka saling berinteraksi satu sama lain.
2.		<i>Initial State</i>	Awal dimulainya suatu aliran kerja.
3.		<i>Final State</i>	Bagian akhir dari suatu aliran kerja.
4.		<i>Decision</i>	Menggambarkan pilihan kondisi yang membuat aliran kerja terbagi menjadi lebih dari satu aliran.
5.		<i>Merge</i>	Menggabungkan kembali aliran kerja yang sebelumnya telah dipecah oleh <i>Decision</i> .
6.		<i>Transition</i>	Menunjukkan aliran proses.

Gambar 2.3 Notasi pada Activity Diagram

3. Class Diagram

Class Diagram menggambarkan struktur fundamental dari sistem. Diagram ini menggambarkan bagaimana kelas objek diorganisir dengan cermat dan


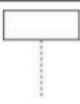
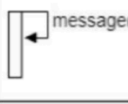

mempertahankan hubungan antar kelas maupun dalam kelas objek untuk mencapai tujuan tertentu.

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
2.		<i>One to Many</i>	Menggambarkan relasi antar tabel yang bersifat <i>one to many</i> .
3.		<i>One to One</i>	Menggambarkan relasi antar tabel yang bersifat <i>one to one</i> .

Gambar 2.4 Notasi pada Class Diagram

4. Sequence Diagram

Sequence Diagram menggambarkan interaksi atau komunikasi antar objek dengan mengirimkan pesan-pesan yang berbeda selama pelaksanaan operasi dalam suatu use case. Diagram ini juga menjelaskan urutan pengiriman dan penerimaan pesan sesuai dengan alur yang telah ditentukan.

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Menggambarkan himpunan peran yang berinteraksi dengan sistem.
2.		<i>Life Line</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
3.		<i>Object Message</i>	Menggambarkan pesan/hubungan antar objek yang menunjukkan urutan kejadian.
4.		<i>Message to Self</i>	Menggambarkan pesan/hubungan antar objek dengan objek itu sendiri
5.		<i>Return Message</i>	Menggambarkan pesan/hubungan antar objek yang menunjukkan urutan kejadian.

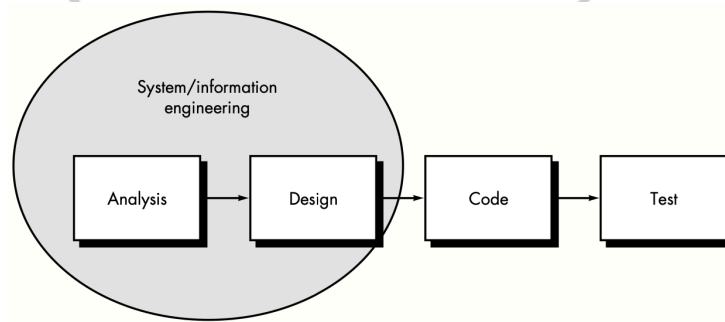
Gambar 2.5 Notasi pada Sequence Diagram

D. Metode Pengembangan

Terdapat beberapa metode pengembangan yang digunakan untuk memfasilitasi proses pengembangan ini (Pressman & Maxim, 2019), yaitu sebagai berikut:

1. Model *Waterfall*

Model *Waterfall* merupakan metode pengembangan perangkat lunak yang klasik, di mana proses pengembangan dibagi menjadi serangkaian fase yang dilaksanakan secara berurutan dan linier. Setiap fase, mulai dari analisis kebutuhan, desain, implementasi, hingga pengujian, harus diselesaikan sepenuhnya sebelum melanjutkan ke fase selanjutnya. Pendekatan ini memungkinkan tim pengembang untuk berkonsentrasi pada satu aspek pada satu waktu, sehingga mempermudah pengelolaan dan pengendalian setiap tahap.



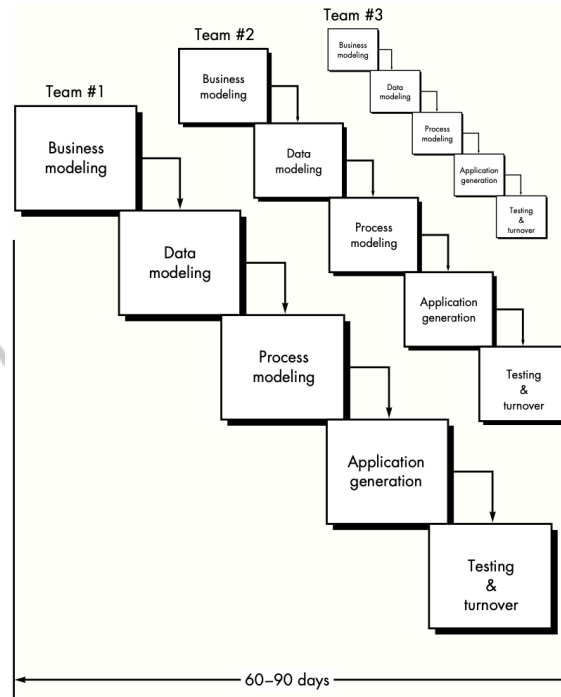
Gambar 2.6 Model Pengembangan *Waterfall*

Kelebihan dari model *Waterfall* adalah kesederhanaannya. Dengan struktur yang jelas, pemangku kepentingan dapat dengan mudah memahami proses dan status proyek. Namun, kelemahan utamanya adalah kurangnya fleksibilitas. Jika ada perubahan kebutuhan setelah fase analisis selesai, akan sulit untuk kembali dan melakukan perubahan. Hal ini berisiko menimbulkan masalah besar jika kesalahan ditemukan di tahap akhir pengembangan.

2. Model RAD (*Rapid Application Development*)

Rapid Application Development (RAD) merupakan model proses pengembangan perangkat lunak yang bersifat inkremental dan menekankan pada siklus pengembangan yang pendek. Model ini merupakan penyederhanaan dari model *waterfall* yang lebih cepat, dengan mengandalkan penggunaan komponen-komponen yang sudah ada. Ketika persyaratan proyek sudah dipahami dengan baik dan ruang lingkup proyek terbatas, proses RAD memungkinkan tim pengembang

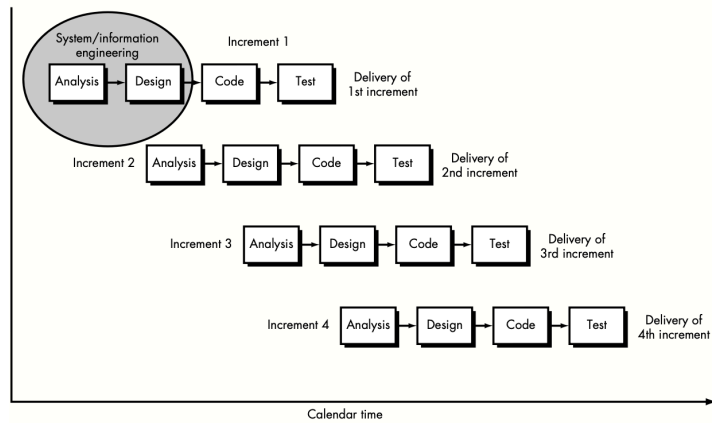
untuk menghasilkan "sistem yang sepenuhnya fungsional" dalam waktu yang relatif singkat, seperti 60 hingga 90 hari.



Gambar 2.7 Model Pengembangan RAD (Rapid Application Development)

3. Model Iteratif dan Inkremental

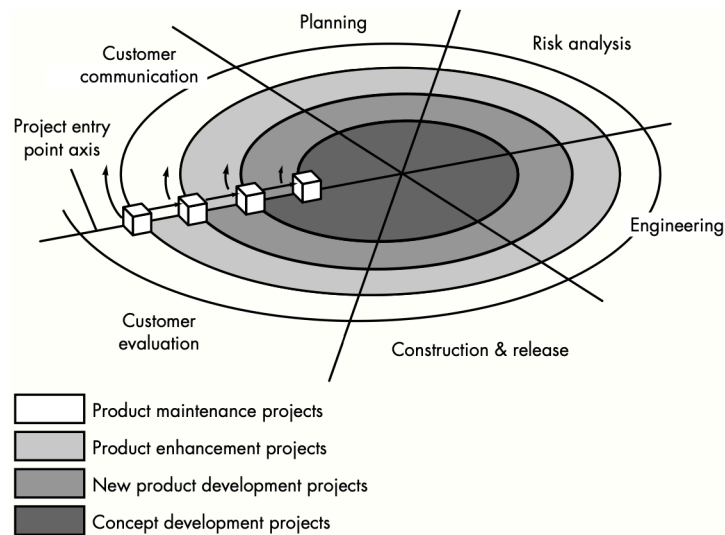
Sebagai respons terhadap keterbatasan model Waterfall, muncul model Iteratif dan Inkremental. Model ini memungkinkan pengembang untuk mengembangkan perangkat lunak secara bertahap melalui iterasi. Setiap iterasi menciptakan versi baru dari perangkat lunak yang mencakup fungsionalitas tambahan. Dengan pendekatan ini, tim dapat menerima umpan balik dari pengguna pada setiap tahap, yang kemudian digunakan untuk memperbaiki dan meningkatkan sistem di iterasi berikutnya.



Gambar 2.8 Model Pengembangan Iteratif dan Inkremental

4. Model Spiral

Model Spiral adalah pendekatan yang menggabungkan elemen desain, pengembangan, dan manajemen risiko. Model ini dibangun di atas prinsip bahwa pengembangan perangkat lunak melibatkan iterasi berulang. Dalam setiap putaran spiral, proyek mengalami beberapa tahap: perencanaan, analisis risiko, pengembangan, dan evaluasi.



Gambar 2.9 Model Pengembangan Spiral

Setiap putaran dimulai dengan perencanaan dan identifikasi tujuan serta risiko yang mungkin muncul. Kemudian, analisis risiko dilakukan untuk mengevaluasi potensi masalah dan mencari solusi. Pengembangan dan pengujian berlangsung secara bersamaan, sehingga tim dapat

menciptakan prototipe yang dapat diuji dan diperbaiki. Dengan menggunakan model ini, organisasi dapat mengelola proyek besar dan kompleks dengan lebih efektif, terutama ketika ketidakpastian tinggi, sehingga memungkinkan untuk melakukan penyesuaian berdasarkan umpan balik yang diperoleh dari setiap iterasi.

2.1.2 Project Management

Menurut Institute (2021), manajemen proyek merupakan penerapan pengetahuan, keterampilan, alat, dan teknik dalam aktivitas proyek untuk mencapai tujuan yang telah ditentukan. Tim proyek dapat memilih untuk menggunakan metode prediktif, campuran, atau adaptif berdasarkan karakteristik proyek tersebut. Proses manajemen proyek diorganisir ke dalam masukan, alat, teknik, dan keluaran yang disesuaikan dengan kebutuhan proyek dan organisasi. Proses ini dapat berulang di setiap tahap siklus proyek dan mencakup lima kelompok utama:

1. **Initiating:** Mendefinisikan dan mendapatkan otorisasi proyek.
2. **Planning:** Menetapkan ruang lingkup dan tujuan proyek.
3. **Executing:** Melaksanakan rencana proyek.
4. **Monitoring and Controlling:** Melacak dan mengatur kemajuan proyek.
5. **Closing:** Menyelesaikan atau menutup proyek atau fase.

Manajemen proyek terbagi ke dalam 10 Knowledge Areas yang mencakup berbagai aspek penting yang perlu dikelola untuk memastikan kelancaran pelaksanaan proyek. Setiap Knowledge Area tersebut terdiri dari serangkaian proses yang saling mendukung dalam menjalankan kegiatan proyek secara efektif (Institute, 2017), sebagai berikut:

1. Manajemen Integrasi Proyek

Manajemen Integrasi Proyek memastikan seluruh elemen proyek berjalan harmonis untuk mencapai tujuan yang ditetapkan. Proses ini mencakup pengembangan *project charter*, penyusunan rencana manajemen proyek, pelaksanaan dan pengelolaan pekerjaan, serta pengelolaan pengetahuan.

2. Manajemen Ruang Lingkup Proyek

Manajemen ruang lingkup proyek memastikan bahwa proyek hanya mencakup pekerjaan yang diperlukan untuk mencapai hasil yang diinginkan. Ini melibatkan perencanaan ruang lingkup, pengumpulan kebutuhan, definisi ruang lingkup, pembuatan struktur perincian pekerjaan (WBS), serta validasi dan pengendalian ruang lingkup agar tetap sesuai dengan rencana.

3. Manajemen Jadwal Proyek

Manajemen Jadwal Proyek mengatur aktivitas agar proyek selesai tepat waktu. Proses ini mencakup perencanaan jadwal, identifikasi dan pengurutan aktivitas, perkiraan durasi, penyusunan jadwal, dan pengendalian jadwal untuk memastikan pelaksanaan proyek sesuai dengan tenggat yang ditetapkan.

4. Manajemen Biaya Proyek

Manajemen Biaya Proyek bertujuan memastikan proyek selesai dalam batas anggaran. Proses ini melibatkan perencanaan biaya, estimasi biaya untuk setiap aktivitas, penentuan anggaran keseluruhan proyek, serta pengendalian biaya selama proyek berlangsung.

5. Manajemen Kualitas Proyek

Manajemen Kualitas Proyek bertujuan untuk memastikan bahwa proyek memenuhi standar kualitas yang telah ditentukan. Proses ini mencakup perencanaan, pengelolaan, dan pengendalian kualitas, serta pengukuran hasil proyek untuk memastikan kesesuaian dengan standar yang berlaku.

6. Manajemen Sumber Daya Proyek

Manajemen Sumber Daya Proyek mengelola sumber daya manusia dan fisik agar proyek berjalan sesuai rencana. Ini mencakup perencanaan dan perolehan sumber daya, pengembangan tim, serta pemantauan dan pengendalian penggunaan sumber daya selama proyek berlangsung.

7. Manajemen Komunikasi Proyek

Manajemen Komunikasi Proyek memastikan informasi proyek dikomunikasikan secara tepat waktu dan jelas kepada semua pemangku kepentingan. Proses ini melibatkan perencanaan, pengelolaan, dan pemantauan komunikasi sepanjang siklus proyek.

8. Manajemen Risiko Proyek

Manajemen Risiko Proyek bertujuan mengidentifikasi, menganalisis, dan merespons risiko yang dapat mempengaruhi proyek. Proses ini mencakup perencanaan manajemen risiko, identifikasi risiko potensial, analisis dampak, serta pengembangan dan pelaksanaan tanggapan risiko.

9. Manajemen Pengadaan Proyek

Manajemen Pengadaan Proyek mengelola proses pengadaan barang atau jasa dari pihak eksternal. Ini melibatkan perencanaan kebutuhan pengadaan, pelaksanaan pengadaan, pengendalian hubungan dengan vendor, serta penutupan kontrak setelah barang atau jasa diterima.

10. Manajemen Pemangku Kepentingan

Manajemen Pemangku Kepentingan dalam proyek berfokus pada pengelolaan partisipasi para pemangku kepentingan. Proses ini melibatkan identifikasi pihak-pihak terkait, penyusunan strategi untuk melibatkan mereka, pengelolaan ekspektasi, serta pemantauan keterlibatan untuk memastikan tercapainya dukungan terhadap tujuan proyek.

Dalam manajemen proyek, terdapat konsep Triple Constraint yang terdiri dari tiga elemen utama: waktu, biaya, dan ruang lingkup. Konsep ini sering digambarkan dalam bentuk segitiga, di mana perubahan pada satu elemen dapat memengaruhi kedua elemen lainnya, yaitu sebagai berikut:

1. **Waktu (*Time*)**

Waktu dalam manajemen proyek merujuk pada durasi yang dibutuhkan untuk menyelesaikan proyek sesuai jadwal. Proses ini melibatkan perencanaan aktivitas, estimasi durasi, pembuatan jadwal, dan pengendalian waktu untuk memastikan proyek selesai tepat waktu.

2. **Biaya (Cost)**

Biaya mencakup seluruh pengeluaran yang diperlukan untuk proyek, termasuk tenaga kerja, bahan, dan peralatan. Manajemen biaya melibatkan perencanaan, estimasi, penentuan anggaran, serta pengendalian pengeluaran agar proyek tetap dalam batas anggaran yang ditetapkan.

3. **Ruang Lingkup (Scope)**

Ruang lingkup proyek mencakup semua pekerjaan yang diperlukan untuk mencapai hasil yang diinginkan. Manajemen ruang lingkup melibatkan perencanaan, definisi ruang lingkup, penyusunan Work Breakdown Structure (WBS), serta pengendalian untuk memastikan proyek tetap sesuai dengan yang direncanakan.

2.1.3 *Database*

Database atau Basis data adalah kumpulan data operasional yang disimpan dan digunakan oleh sistem aplikasi dalam suatu organisasi, yang berbeda dari data input dan output. Data operasional ini digunakan untuk mendukung aktivitas dan proses dalam organisasi tersebut (Narang, 2018).

- *Database Management System (DBMS)* memungkinkan pemusatan data organisasi di satu lokasi, sehingga data dapat dibagikan di antara pengguna, standar dapat diterapkan, duplikasi data dapat dihindari, dan independensi data secara logis maupun fisik dapat tercapai. DBMS dilengkapi dengan berbagai fitur modern seperti *concurrency control*, *automatic recovery*, *backup*, *security*, dan lain lain.
- *Relational Database Management System (RDBMS)* adalah sistem yang mengelola basis data dengan menggunakan model data relasional. Pada RDBMS, data disusun dalam bentuk tabel yang terdiri dari baris dan kolom. Setiap tabel biasanya memiliki kunci utama (*primary key*) yang berfungsi untuk mengidentifikasi setiap baris secara unik, serta kunci asing (*foreign key*) yang digunakan untuk membangun hubungan antar tabel. Relasi antar tabel dalam Database, antara lain:

- a. One-to-One
 - b. One-to-Many
 - c. Many-to-Many
- Normalisasi *Database* merupakan proses dalam model data relasional yang bertujuan untuk mengorganisasi data dengan ketergantungan yang erat. Proses ini dilakukan untuk mencegah duplikasi tabel dalam basis data dengan cara mendekomposisi tabel yang masih mengandung anomali atau ketidakwajaran. Hasilnya adalah tabel-tabel yang lebih sederhana dan terstruktur, tanpa adanya redundansi data, sehingga memungkinkan pengguna untuk melakukan operasi insert, delete, dan update pada baris (record) tanpa menimbulkan inkonsistensi data.
 - SQL (*Structured Query Language*) merupakan bahasa pemrograman baku yang digunakan untuk mengelola serta memodifikasi basis data relasional (Efendy, 2018). *Structured Query Language* (SQL) terdiri dari tiga kelompok pernyataan berdasarkan fungsinya:
 - a. *Data Definition Language* (DDL), yang digunakan untuk menentukan tipe data, hubungan antar data, dan validasi, melalui perintah-perintah seperti *create*, *drop*, dan *alter*
 - b. *Data Manipulation Language* (DML), yang berfungsi untuk memilih dan memanipulasi data, dengan menggunakan perintah-perintah seperti *select*, *update*, *insert*, dan *delete*
 - c. *Data Control Language* (DCL), yang bertanggung jawab untuk mengelola hak akses dan kontrol terhadap data, dengan perintah seperti *grant*.

2.1.4 Teknik Pengumpulan Data

Metode pengumpulan data merujuk pada cara atau pendekatan yang digunakan untuk mengumpulkan informasi yang akan dianalisis. Dengan demikian, teknik ini membutuhkan prosedur yang terencana dan terstruktur guna memperoleh data yang akurat dan sesuai dengan fakta yang ada. Terdapat beberapa teknik yang dapat dilakukan dalam mengumpulkan data:

1. Studi Literatur

Sekumpulan aktivitas yang terkait dengan teknik pengumpulan data pustaka, seperti membaca, mencatat, dan mengelola bahan penelitian (Kartiningrum, 2015). Oleh karena itu, informasi yang digunakan sebagai referensi berasal dari buku, jurnal, dan dokumen lainnya yang telah dipublikasikan.

2. Wawancara

Metode wawancara adalah suatu situasi interaksi langsung antara pewawancara dan responden yang bertujuan untuk memperoleh informasi yang dibutuhkan, serta untuk mendapatkan data mengenai responden dengan meminimalkan bias dan mengoptimalkan efisiensi (Hakim, 2013:167).

3. Observasi

Metode observatif adalah suatu pendekatan yang dilakukan melalui pengamatan ilmiah dengan memanfaatkan indra untuk menarik kesimpulan mengenai hubungan, sebab-akibat, serta makna dari suatu situasi (Abdullah et al., 2022:26).

4. Kuisisioner

Kuesioner merupakan metode pengumpulan data di mana responden memberikan jawaban atas serangkaian pertanyaan tertulis yang disusun dengan cara yang terstruktur. Metode ini digunakan untuk memperoleh informasi langsung dari responden mengenai pandangan, opini, serta pengalaman mereka terkait dengan variabel yang sedang diteliti (Sugiyono, 2021).

2.1.5 *Black-Box Testing*

Menurut Pressman dan Maxim (2019), pengujian black-box memfokuskan perhatian pada aspek fungsional perangkat lunak. Hal ini berarti bahwa pengujian jenis ini memungkinkan insinyur perangkat lunak untuk menentukan rangkaian kondisi input yang akan menguji secara menyeluruh seluruh kebutuhan fungsional suatu program. Pengujian black-box tidak

dimaksudkan sebagai pengganti teknik white-box, melainkan sebagai pendekatan yang saling melengkapi, yang cenderung mengidentifikasi jenis kesalahan yang berbeda. Pengujian black-box bertujuan untuk mendeteksi kesalahan dalam kategori berikut:

1. Fungsi yang tidak tepat atau tidak ada.
2. Kesalahan pada antarmuka pengguna.
3. Kesalahan dalam struktur data atau akses ke basis data eksternal.
4. Kesalahan dalam perilaku atau kinerja.
5. Kesalahan pada proses inisialisasi dan terminasi.

2.2 Tinjauan Studi

1. Jurnal **SISFOKOM (Sistem Informasi Dan Komputer)**, Vol. 9, No. 3, halaman 365 – 372 memuat penelitian yang dilakukan oleh Doni Darmawan, Anita Ratnasari dari Universitas Mercu Buana, pada tahun 2020 dengan judul **“RANCANG BANGUN SISTEM INFORMASI MANAJEMEN PROYEK BERBASIS WEB PADA PT SEATECH INFOSYS”**. Peneliti mengevaluasi efektivitas penerapan aplikasi manajemen proyek di perusahaan konsultan IT, dengan fokus pada PT. Seatech Infosys. Temuan penelitian ini mengungkapkan bahwa penggunaan alat sederhana seperti Microsoft Excel dalam pengelolaan proyek menimbulkan berbagai masalah, termasuk kesalahan dalam pencatatan, kesulitan berbagi informasi, serta pengelolaan proyek yang tidak terstruktur. Dengan mengadopsi metode waterfall, dikembangkanlah sebuah Sistem Informasi Manajemen Proyek berbasis web yang terbukti efektif mengatasi masalah tersebut. Sistem ini memungkinkan manajemen untuk memonitor ketersediaan tenaga ahli, perkembangan proyek, dan pengeluaran dengan lebih terorganisir dan efisien. Hasil pengujian menunjukkan bahwa sistem ini berhasil mendukung pengelolaan sumber daya, waktu, dan biaya secara efektif.
2. Jurnal **Pengembangan Ilmu Komputer dan Teknologi Informasi**, Vol. 4, No. 6, halaman 1819 – 1827 memuat penelitian yang dilakukan oleh Arif

Hendrawan dan Nurudin Santoso dari Universitas Brawijaya, pada tahun 2020 dengan judul **“RANCANG BANGUN SISTEM INFORMASI MANAJEMEN PROYEK KOLABORASI KERJA BERBASIS MOBILE (STUDI KASUS : DEBOX INDONESIA)”**. Penelitian ini berfokus pada pengembangan sistem manajemen proyek berbasis mobile menggunakan Agile Project Management dengan Scrum di perusahaan Debox Indonesia. Sistem ini dirancang untuk menggantikan aplikasi *spreadsheet* yang tidak memadai, memungkinkan manajemen proyek yang lebih efisien dan akurat. Dengan menggunakan framework React Native dan GraphQL, sistem ini mendukung fungsionalitas utama seperti scrumming proyek, pelaporan, dan kolaborasi antar karyawan. Hasilnya, sistem ini mencatat 67 kebutuhan fungsional dan telah diuji secara menyeluruh, membuktikan validitasnya dan kompatibilitas dengan berbagai perangkat, sehingga meningkatkan efektivitas manajemen proyek di lingkungan kerja yang dinamis.

3. Jurnal **Seminar Nasional Informatika – FTI UPGRIS**, Vol. 2, memuat penelitian yang dilakukan oleh Hamzah Noer Arifin, Nur Latifah Dwi Mutiara Sari dari Universitas PGRI Semarang, pada tahun 2024 berjudul **“RANCANG BANGUN SISTEM INFORMASI MANAJEMEN PROYEK BERBASIS WEB PADA PT. ABISEKA.”** Penelitian ini berpusat pada pembuatan sistem informasi manajemen proyek berbasis web di PT. ABISEKA. Sistem tersebut dikembangkan menggunakan metodologi Rapid Application Development (RAD), yang dikenal mampu mempercepat proses pengembangan dengan pendekatan berbasis komponen. Penelitian ini berhasil menunjukkan bahwa sistem yang dikembangkan mampu memfasilitasi komunikasi, kolaborasi, dan pemantauan kemajuan proyek secara lebih efisien, dengan fitur-fitur utama seperti pengelolaan data proyek, obrolan antar pengguna, dan laporan perkembangan proyek yang akurat. Hasil penelitian ini memberikan kontribusi signifikan berupa model sistem yang dapat diadaptasi oleh perusahaan lain untuk meningkatkan efisiensi manajemen proyek mereka.

4. Dalam *International Journal of Recent Technology and Engineering (IJRTE)*, Vol. 8, No. 6, halaman 5282 – 5286 memuat penelitian yang dilakukan oleh M. Suresh dkk pada tahun 2020 dengan judul **“DEVELOPMENT OF WORKLOAD DISTRIBUTION SYSTEM USING RANDOM SAMPLE PARTITION ALGORITHM”** mengkaji pengembangan sistem manajemen beban kerja yang bertujuan untuk mendistribusikan tugas secara adil di antara anggota tim dengan menganalisis data terkait tugas dan kompleksitasnya. Penelitian ini mengimplementasikan algoritma partisi sampel acak untuk meminimalkan beban analisis dengan membagi data menjadi blok-blok yang lebih sederhana, yang memungkinkan analisis yang lebih efisien dan akurat. Sistem ini menyajikan hasil analisis secara jelas kepada manajer, membantu dalam pengalokasian tugas baru secara merata dan menghindari penumpukan beban kerja pada karyawan tertentu. Implementasi sistem ini menunjukkan bahwa otomatisasi dapat meningkatkan efisiensi manajerial dan mengurangi ketidakadilan dalam distribusi tugas, serta meningkatkan keseluruhan kinerja tim melalui analisis yang lebih baik.
5. Dalam *Journal of Emerging Information Systems and Business Intelligence (JEISBI)*, Vol. 03, No. 04, Tahun 2022, penelitian berjudul **“Rancang Bangun Sistem Informasi Manajemen Proyek Berbasis Website Menggunakan *Project Management Body of Knowledge 6 (Studi Kasus PT. Tekno Mandala Kreatif)*”** yang dilakukan oleh Alfando Vifan Suwandana dan Ardhini Warih Utami dari Universitas Negeri Surabaya membahas pengembangan sistem informasi manajemen proyek berbasis web. Penelitian ini menggunakan lima area pengetahuan dari PMBOK edisi ke-6, yaitu manajemen integrasi, ruang lingkup, jadwal, biaya, dan pemangku kepentingan proyek. Sistem ini dirancang untuk membantu Project Manager dalam pengelolaan proyek, termasuk monitoring, dokumentasi, dan pelaporan, yang sebelumnya dilakukan secara tidak terstruktur. Implementasi sistem diharapkan dapat meningkatkan keakuratan pengawasan proyek dan mendukung efisiensi dalam proses

manajemen proyek. Pengembangan sistem dilakukan dengan menerapkan metode Agile melalui pendekatan Extreme Programming, yang mencakup tahap perencanaan, desain, implementasi, serta pengujian menggunakan metode black-box testing. Hasil penelitian membuktikan bahwa sistem yang dibangun mampu beroperasi dengan baik sesuai kebutuhan pengguna, mendukung pengelolaan proyek yang lebih terstruktur, dan membantu proses pengambilan keputusan di PT. Tekno Mandala Kreatif.

