BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian tentang penerapan dan pengembangan sistem SCADA (*Supervisory Control and Data Acquisition*) telah banyak dilakukan dan memberikan penjelasan tentang manfaat serta penerapannya di berbagai sektor industri. Tabel 2.1 berikut ini merangkum beberapa penelitian terdahulu yang sesuai dengan pengembangan dan penerapan sistem SCADA dalam lima tahun terakhir. Melalui penelitian-penelitian ini, terlihat jelas bagaimana sistem SCADA dalam implementasinya dapat memberikan manfaat bagi perusahaan dalam mengoptimalkan proses kerja mereka.

Table 2.1 Penelitian Terdahulu

No	Judul Penelitian	Nama Jurnal, Vol, Edisi, & Tahun	Penulis	Masalah	Metode	Hasil
1	Sistem Monitoring Kinerja Sub Bagian Operator SCADA Menggunakan *Agile* Development Methods	Jurnal Sistem Informasi, Vol. 8, No. 1, 2021	Indah Kusuma Dewi, Randa Amalta Saputra / 2021	Monitoring kinerja operator SCADA masih manual dan kurang efisien	*Agile* Development, PHP, MySQL	Sistem monitoring berbasis web meningkatkan efisiensi kerja operator SCADA dan mempermudah manajemen
2	Implementasi SCADA Pada Monitoring Penggunaan Energi Listrik di Gedung Teknik Elektro Universitas	Jurnal Teknologi Energi, Vol. 5, No. 2, 2022	Richard John Octavianus S, Subairi, Rifki Hari Romadhon / 2024	Pemantauan energi listrik manual menyebabkan inefisiensi dan kurangnya penghematan energi.	Modbus, LoRaWAN	SCADA berbasis Modbus dan LoRaWAN memungkinkan pemantauan energi secara *real-time*, meningkatkan potensi penghematan energi di fasilitas pendidikan.
3	Sistem Monitoring Instrument Air Compressor (IAC)	Jurnal Teknologi Industri, Vol. 7, No. 3, 2021	Muchamad Chadiq Zakaria, Edy Kurniawan, Jawwad	Monitoring kinerja kompresor udara tidak *real-time* dan sulit	Modbus RTU RS-485	Sistem monitoring SCADA memungkinkan pemantauan *real-time* kinerja IAC

	Berbasis SCADA dengan Komunikasi Modbus RTU RS-485		Sulthon H / 2021	diintegrasikan dengan sistem kontrol lain.		dan meningkatkan keandalan dalam lingkungan industri.
4	Implementasi Web SCADA Pada Sistem PLTS	Jurnal Energi Terbarukan, Vol. 6, No. 1, 2023	Rose Sunky, Riki Mukhaiyar / 2023	Sistem PLTS belum menggunakan pemantauan jarak jauh yang efisien	Web SCADA, Internet, Sensor	SCADA berbasis web memudahkan pemantauan PLTS secara jarak jauh dan mendukung pengelolaan energi terbarukan dengan lebih efisien.
5	Implementasi dan Perkembangan Sistem SCADA di Industri: Tinjauan dari Sudut Pandang Pakar	Jurnal Teknik Sipil, Vol. 12, No. 3, 2022	David Fauzan, Gresia Febriana Tambunan, Stefanus Ivan Bohal Siringoringo / 2023	SCADA perlu dioptimalkan dengan integrasi teknologi baru untuk menghadapi tantangan industri 4.0.	Tinjauan literatur, Wawancara Pakar	SCADA berkembang dengan integrasi IoT dan Cloud Computing, meningkatkan efisiensi, namun menghadapi tantangan keamanan data.
6	Perancangan SCADA pada Distribution Station dan Pick & Place Station Menggunakan Metode *Waterfall*	e-Proceeding of Engineering: Vol. 8, No. 5, Oktober 2021	Evelin Sekar Dewinta, Haris Rachmat, Denny Sukma Eka Atmaja / 2021	Pengoperasian SCADA pada simulator bottling plant belum optimal karena tidak terintegrasi dengan *database*	Metode *Waterfall*, HMI, *Database*	Sistem SCADA berhasil mengintegrasikan HMI dan *database*, memudahkan pelaporan dan pemantauan stasiun Distribution serta Pick & Place.
7	Sistem Informasi Monitoring dan Pemeliharaan Penggunaan SCADA	Jurnal TEKNO KOMPAK, Vol. 15, No. 2, 2024	Arief Budiman, Jupriyadi, Sunariyo / 2024	Pemeliharaan alat SCADA di PT PLN dilakukan secara manual, tidak efisien, dan menyulitkan pelacakan data	Web Engineering, CodeIgniter, PHP, MySQL	Sistem monitoring berbasis web dengan framework CodeIgniter mempercepat proses pemeliharaan SCADA dan lebih efisien dalam penyimpanan serta pelacakan data.
8	Penerapan Teknologi SCADA pada Sistem Distribusi Air Bersih	Jurnal Teknik Sipil, Vol. 12, No. 3, 2022	Andika Pratama, Liana Novita	Monitoring distribusi air bersih masih dilakukan secara manual, sehingga menyebabkan kesulitan dalam	SCADA, IoT, Sensor	Penerapan sistem SCADA pada distribusi air bersih berhasil meningkatkan efisiensi pemantauan dan pengelolaan sumber daya air,

				pengelolaan dan kontrol sistem.		serta memungkinkan kontrol yang lebih baik terhadap aliran dan kualitas air.
9	Implementation of Total Productive *Maintenance* in Manufacturing Industries: A Case Study	Procedia Manufacturing (Vol 30, 2019)	Ahuja, I.P.S., Khamba, J.S.	Rendahnya efektivitas mesin dan tingginya *Downtime* dalam industri manufaktur	Penerapan Total Productive *Maintenance* (TPM) dan pengukuran Overall Equipment Effectiveness (OEE)	Peningkatan OEE sebesar 15% dan penurunan *Downtime* mesin sebesar 20% setelah implementasi TPM
10	Reducing *Downtime* and Increasing Productivity through Total Productive *Maintenance*	International Journal of Production Research (Vol 52, No 1, 2014)	Muchiri, P., Pintelon, L.	Tingginya *Downtime* mesin yang mengakibatkan penurunan produktivitas	Analisis six big losses dan penerapan TPM untuk mengidentifikasi dan mengurangi sumber *Downtime*	Penurunan *Downtime* sebesar 25% dan peningkatan produktivitas sebesar 10% setelah identifikasi dan pengurangan six big losses
11	Application of Lean Manufacturing *Tools* in Reducing Assembly Line Waste in an Automotive Industry	International Journal of Engineering Research and Applications (Vol 5, No 8, 2015)	Vinodh, S., Arvind, K.R., Somanaathan, M.	Pemborosan waktu dan sumber daya pada lini perakitan industri otomotif	Penerapan alat Lean Manufacturing seperti 5S, Kaizen, dan Value Stream Mapping untuk mengidentifikasi dan mengurangi pemborosan	Pengurangan waktu siklus sebesar 12% dan peningkatan efisiensi lini perakitan sebesar 18% setelah implementasi Lean Manufacturing
12	*Development and Design of Internship Search Information System Website for Multimedia Nusantara University Students*	International Conference on Entrepreneurship and Business Management (ICEBM 2013), Bali, 2013	Enrico Siswanto, Johan Setiawan	Kurangnya sistem penghubung antara mahasiswa dan Career Development UMN dalam pencarian tempat magang	Model *Waterfall* (analisis data, desain prototipe, dan implementasi sistem)	Website yang dirancang memfasilitasi mahasiswa dalam mencari informasi magang, memungkinkan perusahaan langsung memposting lowongan magang, serta meningkatkan layanan Career Development kepada mahasiswa.
13	*The Development of Web-based Sales and Inventory*	Ultima Infosys, Vol. 14, No. 2, 2023	Chyntia Cahya Utami, Ririn Ikana Desanti,	Masalah pencatatan transaksi yang masih manual, sehingga tidak	Metode prototipe	Sistem berbasis web yang terintegrasi berhasil dikembangkan

	System for a Stationary Store		Fransiscus Ati Hatim	efisien dan sulit mengetahui stok serta laba/rugi.		untuk mencatat data transaksi dan stok secara otomatis serta menghasilkan laporan secara efisien.

Penelitian terdahulu menunjukkan relevansi dalam meningkatkan efisiensi operasional melalui penerapan teknologi seperti *Total Productive Maintenance* (TPM) dan *Lean Manufacturing* untuk mengurangi *Downtime* serta pemborosan. Korelasinya terletak pada tujuan yang sama, yaitu meningkatkan produktivitas dan efisiensi melalui teknologi, yang menjadi dasar bagi penelitian ini. Namun, pendekatan penelitian sebelumnya hanya terbatas pada pemantauan operasional umum atau pengurangan *Downtime* tanpa integrasi sistem yang lengkap. Sebaliknya, penelitian yang akan dikembangkan menonjolkan integrasi modul penting seperti pencatatan *Downtime*, pengajuan permintaan *spare part* berbasis web, dan sistem pemantauan secara terpusat untuk mengoptimalkan efisiensi operasional dan manajemen *maintenance* secara menyeluruh.

Pada tabel 2.1 penelitian terdahulu, terdapat beberapa research gap dengan penelitian yang akan dibangun, terutama dalam hal integrasi dan fokus pada efisiensi operasional *maintenance* secara menyeluruh. Penelitian-penelitian sebelumnya hanya berfokus pada aspek pengawasan operasional umum, seperti pemantauan energi atau kontrol jarak jauh, namun belum mengintegrasikan modul yang secara khusus mengoptimalkan efisiensi bagian *maintenance*, seperti pencatatan otomatis *Downtime* atau pengajuan permintaan *spare part* untuk perawatan mesin [9]. Sebaliknya, penelitian yang akan dikembangkan memiliki modul khusus yang mencakup pencatatan *Downtime*, pencatatan kebutuhan *spare part*, dan sistem pengawasan berbasis web sehingga lebih mendalam dalam meningkatkan efisiensi kerja bagian *maintenance* secara keseluruhan.

Selain itu, meskipun beberapa penelitian terdahulu telah menerapkan SCADA berbasis web untuk tujuan pemantauan jarak jauh di sektor-sektor tertentu, tidak ada fitur yang memungkinkan integrasi antara pemantauan jarak jauh dengan

pengelolaan operasional *maintenance* dan analisis kinerja *maintenance* yang mendetail [8]. Penelitian yang dibangun kali ini memiliki nilai tambah dalam aspek tersebut, yakni mencakup fitur otomatisasi untuk mencatat *Downtime* secara *real-time* dan mempercepat proses persetujuan *spare part* yang dibutuhkan dalam perawatan mesin, yang pada akhirnya dapat mendukung pencapaian efisiensi operasional perusahaan [8]. Selain itu, penelitian-penelitian sebelumnya cenderung tidak menyediakan sistem monitoring dan kontrol *spare part* secara terintegrasi dalam platform yang sama, padahal kontrol pemeliharaan yang efisien sangat diperlukan untuk menekan risiko keterlambatan dan kesalahan yang mungkin terjadi pada proses pemesanan [10].

2.2 Tinjauan Teori

2.2.1 *Supervisory Control and Data Acquisition*

Supervisory Control and Data Acquisition (SCADA) adalah sistem pengendalian yang digunakan untuk memantau dan mengontrol proses industri dari jarak jauh (Nathan, 2023). Sistem ini mengintegrasikan *Hardware* dan *Software* yang berfungsi untuk mengumpulkan data, memprosesnya, serta mengambil keputusan secara *real-time*. SCADA sangat penting dalam manajemen infrastruktur karena kemampuannya untuk mengotomatisasi berbagai proses industri yang kompleks dan tersebar luas secara geografis [11]. Fungsi utama SCADA adalah untuk memberikan akses terhadap data proses secara *real-time* kepada operator atau sistem kontrol yang terpusat, sehingga keputusan yang tepat dapat diambil dengan cepat.

Misalnya, dalam distribusi energi listrik, SCADA dapat mendeteksi pemadaman listrik dan mengambil langkah-langkah untuk mengalihkan jalur distribusi agar listrik dapat terus disalurkan ke konsumen. Selain itu, SCADA juga dapat digunakan untuk memonitor kinerja mesin dalam industri manufaktur sehingga operator dapat melakukan pemeliharaan preventif sebelum kerusakan terjadi, yang secara langsung berdampak pada efisiensi operasional. Keunggulan SCADA terletak pada skalabilitas dan fleksibilitasnya. Sistem ini dapat

dikustomisasi untuk berbagai jenis aplikasi dan dapat diimplementasikan baik pada skala kecil maupun besar [12].

SCADA juga mendukung berbagai jenis protokol komunikasi, sehingga memudahkan integrasi dengan teknologi yang sudah ada. SCADA sering digunakan dalam infrastruktur utama, serangan siber pada sistem ini dapat memiliki dampak yang sangat besar. Untuk mengatasi masalah ini, banyak perusahaan yang mulai menerapkan langkah-langkah keamanan yang lebih ketat, seperti enkripsi data dan sistem autentikasi ganda [13].

2.2.2 *Downtime* Mesin

Downtime mesin adalah periode ketika sebuah mesin tidak beroperasi atau tidak dapat digunakan karena berbagai alasan, termasuk perawatan, kegagalan teknis, atau penyesuaian proses [14]. *Downtime* mesin merupakan salah satu aspek penting dalam manajemen operasi karena memiliki dampak langsung terhadap produktivitas, biaya, dan efisiensi operasional suatu perusahaan [18]. Dalam sebuah perusahaan, *Downtime* sering kali menjadi indikator utama untuk mengukur seberapa baik kinerja mesin dan proses produksi dikelola. *Downtime* yang tidak terduga dapat menyebabkan penurunan output produksi dan peningkatan biaya perbaikan serta perawatan [15]. Secara umum, *Downtime* mesin dapat dikategorikan menjadi dua jenis utama, yaitu *Downtime* terencana dan *Downtime* tidak terencana [4].

Downtime terencana mencakup waktu yang dialokasikan untuk perawatan preventif, kalibrasi, atau penggantian suku cadang yang sudah dijadwalkan. Ini penting untuk memastikan bahwa mesin tetap berfungsi dengan baik dan menghindari kerusakan yang lebih serius di masa depan. Sebaliknya, *Downtime* tidak terencana terjadi ketika mesin tiba-tiba berhenti beroperasi karena kerusakan atau kegagalan komponen. *Downtime* tidak terencana sering kali lebih merugikan karena memerlukan tindakan perbaikan segera yang dapat mengganggu jadwal produksi [6]. Salah satu metode yang digunakan untuk mengurangi *Downtime* mesin adalah dengan penerapan program perawatan prediktif dan preventif. Perawatan prediktif melibatkan pemantauan kondisi mesin secara *real-time*

menggunakan sensor dan teknologi analitik untuk mendeteksi tanda-tanda awal kerusakan.

Dampak *Downtime* mesin tidak hanya terbatas pada produktivitas, tetapi juga pada kualitas produk dan keselamatan kerja. Ketika *Downtime* terjadi, proses produksi terhenti, yang dapat menyebabkan keterlambatan dalam pengiriman produk ke pelanggan dan merusak reputasi perusahaan. Selain itu, *Downtime* juga dapat memengaruhi keselamatan kerja, terutama jika mesin mengalami kerusakan yang menyebabkan kondisi kerja menjadi tidak aman. Oleh karena itu, mengelola *Downtime* dengan baik adalah hal yang sangat penting untuk menjaga kelangsungan operasional yang efisien dan aman [10]. Untuk mengurangi *Downtime* mesin, perusahaan perlu menerapkan manajemen perawatan yang efektif, termasuk melakukan audit perawatan secara berkala, melatih karyawan, dan menggunakan alat pemantauan modern. Selain itu, melibatkan tim manajemen dan teknisi dalam perencanaan dan implementasi program perawatan juga sangat penting.

2.2.3 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah seperangkat teknik diagram standar yang bertujuan untuk menyediakan kosa kata umum terkait istilah berorientasi objek dan teknik diagram yang cukup kompleks untuk memodelkan setiap proyek pengembangan sistem, mulai dari tahap analisis hingga implementasi [10]. UML digunakan untuk menggambarkan berbagai komponen sistem *software* secara grafis, yang memudahkan dalam memahami struktur dan perilaku sistem yang sedang dibangun [10]. UML menyediakan satu set teknik diagram yang dapat digunakan untuk memodelkan sistem dari berbagai perspektif, baik dalam hal struktur statis, alur proses dinamis, interaksi antar komponen, serta hubungan objek [10].

UML memiliki 15 jenis diagram yang dibagi dalam dua kategori utama, yaitu diagram struktur dan diagram perilaku [10]. Diagram struktur digunakan untuk memodelkan elemen-elemen statis dalam sistem, seperti objek, kelas, komponen, dan hubungan antar bagian sistem. Diagram ini membantu menggambarkan

bagaimana data diorganisir, disimpan, dan diakses dalam sebuah sistem. Beberapa diagram struktur yang paling umum digunakan dalam UML adalah diagram kelas (*class diagram*), diagram objek (*object diagram*), diagram paket (*package diagram*), diagram komponen (*component diagram*), dan diagram penyebaran (*deployment diagram*). Setiap diagram ini memiliki tujuan yang berbeda, misalnya diagram kelas menggambarkan struktur kelas beserta atribut dan operasi yang dimiliki, sedangkan diagram penyebaran memperlihatkan bagaimana komponen perangkat lunak disebarkan pada perangkat keras dalam infrastruktur yang lebih besar [10].

Selain itu, UML juga menyediakan diagram perilaku yang menggambarkan bagaimana sistem berfungsi secara dinamis. Diagram perilaku ini menggambarkan interaksi antar objek, bagaimana objek berubah status atau kondisi berdasarkan peristiwa tertentu, serta bagaimana proses dalam sistem berjalan dari waktu ke waktu. Beberapa diagram perilaku dalam UML termasuk diagram aktivitas (*activity diagram*), diagram urutan (*sequence diagram*), diagram komunikasi (*communication diagram*), diagram status (*state machine diagram*), dan diagram use case (*use case diagram*). Diagram aktivitas, misalnya, digunakan untuk menggambarkan alur kerja atau serangkaian aktivitas yang terjadi dalam sistem, sementara diagram urutan menggambarkan interaksi antara objek-objek dalam sistem berdasarkan urutan waktu. Diagram *use case* sangat berguna untuk menggambarkan fungsionalitas sistem dari perspektif pengguna, memodelkan kebutuhan sistem berdasarkan peran aktor yang terlibat. UML sangat berguna dalam analisis sistem dan desain perangkat lunak karena memungkinkan pengembang dan analis untuk memvisualisasikan sistem secara jelas dan terstruktur.

2.3 *Framework*

2.3.1 *Waterfall*

Metode pengembangan *Waterfall* adalah pendekatan linier dan sekuensial dalam proses pengembangan sistem yang dikembangkan oleh Dr. Winston W. Royce pada tahun 1970 [9]. Model ini terdiri dari beberapa tahapan yang harus

diselesaikan secara berurutan, dimulai dari analisis kebutuhan, desain sistem, pengembangan, pengujian, hingga pemeliharaan. Setiap tahapan memiliki output yang jelas, dan hasil dari setiap tahap menjadi dasar untuk tahap berikutnya. Pendekatan ini sangat berguna untuk proyek dengan kebutuhan yang sudah ditentukan dengan baik dan tidak banyak berubah, seperti sistem informasi yang kompleks [9].

Kelebihan dari metode *Waterfall* adalah kemampuannya untuk memfasilitasi dokumentasi yang lengkap di setiap tahap, sehingga memudahkan tim untuk melacak kemajuan dan referensi di masa mendatang [16]. Dokumentasi yang baik juga membantu dalam pelatihan dan transfer pengetahuan bagi anggota tim baru. Selain itu, *Waterfall* memungkinkan pemangku kepentingan untuk memiliki pemahaman yang jelas mengenai proses pengembangan dan waktu yang dibutuhkan untuk menyelesaikan proyek, sehingga memudahkan dalam pengelolaan ekspektasi [17].

Namun, metode ini juga memiliki beberapa kelemahan. Salah satu kelemahannya adalah kurangnya fleksibilitas dalam menangani perubahan kebutuhan di tengah proses pengembangan [18]. Jika ada perubahan signifikan yang diperlukan setelah tahap analisis kebutuhan selesai, hal ini dapat memerlukan revisi besar dan mengakibatkan keterlambatan. Meski demikian, dalam konteks sistem kritis seperti SCADA, di mana stabilitas dan keamanan sangat penting, pendekatan *Waterfall* sering kali menjadi pilihan yang lebih baik.

2.3.2 *V-Model*

V-Model, atau Verification and Validation Model, adalah pendekatan pengembangan sistem yang merupakan perpanjangan dari model Waterfall dengan penekanan khusus pada pengujian di setiap tahap pengembangan. Dalam V-Model, setiap fase pengembangan dihubungkan dengan fase pengujian paralel yang sesuai, menciptakan bentuk seperti huruf "V". Model ini dirancang untuk memastikan bahwa setiap tahap, mulai dari analisis kebutuhan hingga implementasi, memiliki validasi yang ketat guna mengidentifikasi dan memperbaiki kesalahan sejak dini.

Tahapan V-Model dimulai dengan analisis kebutuhan, di mana kebutuhan sistem dirumuskan dengan jelas dan menjadi dasar untuk seluruh pengembangan. Selanjutnya, pada tahap desain sistem, arsitektur perangkat lunak dan perangkat keras dirancang untuk memenuhi kebutuhan yang telah ditentukan. Setelah itu, tahap desain detail dilakukan untuk memetakan elemen-elemen spesifik dari sistem, seperti modul dan algoritma. Pada setiap tahap ini, pengujian terkait seperti pengujian persyaratan, pengujian desain, dan pengujian unit telah direncanakan sejak awal untuk memastikan hasil dari setiap tahap sesuai dengan kebutuhan yang telah dirumuskan.

Keunggulan utama V-Model adalah kemampuannya untuk menemukan kesalahan lebih awal dalam siklus pengembangan melalui validasi yang terus-menerus. Hal ini sangat penting untuk proyek-proyek yang memerlukan akurasi tinggi, seperti sistem kritis di sektor medis, transportasi, atau keamanan. Dengan memvalidasi setiap langkah, risiko kesalahan di tahap akhir, seperti dalam model Waterfall, dapat diminimalkan. Model ini juga memastikan kualitas sistem yang lebih tinggi, karena pengujian terintegrasi langsung dengan proses pengembangan.

Namun, V-Model memiliki beberapa keterbatasan. Sama seperti Waterfall, V-Model tidak fleksibel terhadap perubahan kebutuhan di tengah proses pengembangan. Jika terjadi perubahan, seluruh siklus pengembangan mungkin harus diulang untuk mengakomodasi kebutuhan baru, yang memakan waktu dan biaya tambahan. Selain itu, pengujian yang intensif di setiap tahap membutuhkan sumber daya yang besar, baik dalam hal tenaga kerja maupun alat, yang dapat meningkatkan biaya proyek secara keseluruhan.

Dalam aplikasi praktis, V-Model sangat cocok untuk proyek yang memiliki persyaratan tetap dan tidak berubah, seperti sistem Supervisory Control and Data Acquisition (SCADA) atau aplikasi perangkat lunak kritis lainnya. Namun, model ini kurang ideal untuk proyek yang menghadapi kebutuhan yang dinamis atau belum terdefinisi dengan jelas. Meskipun demikian, V-Model tetap menjadi pilihan utama untuk proyek yang memprioritaskan kualitas, keandalan, dan validasi yang ketat di setiap tahapan pengembangannya.

2.3.3 Incremental Model

Incremental Model adalah pendekatan pengembangan sistem yang membagi proyek menjadi beberapa iterasi atau modul kecil yang dapat dikembangkan secara bertahap. Dalam model ini, fungsionalitas dasar sistem dirancang dan diimplementasikan terlebih dahulu, sementara fitur tambahan dikembangkan dalam siklus iterasi berikutnya. Setiap iterasi mencakup tahapan seperti analisis kebutuhan, desain, implementasi, dan pengujian. Pendekatan ini memungkinkan sistem untuk digunakan lebih awal, meskipun belum sepenuhnya selesai.

Tahapan Incremental Model dimulai dengan analisis kebutuhan untuk menentukan prioritas modul yang paling penting. Modul pertama biasanya mencakup fungsi inti yang sangat diperlukan untuk operasi dasar. Setelah modul ini selesai, modul-modul berikutnya dikembangkan dengan menambahkan fitur tambahan atau memperluas fungsionalitas sistem. Dalam setiap iterasi, pengujian dilakukan untuk memastikan bahwa modul yang baru dikembangkan bekerja dengan baik dan dapat diintegrasikan dengan modul yang sudah ada.

Keunggulan utama Incremental Model adalah fleksibilitasnya. Dengan model ini, pengembang dapat menangani kebutuhan yang berubah selama proses pengembangan dengan memasukkan perubahan tersebut ke dalam iterasi berikutnya. Hal ini menjadikan Incremental Model sangat cocok untuk proyek dengan kebutuhan yang dinamis atau belum sepenuhnya terdefinisi di awal. Selain itu, pendekatan bertahap ini memungkinkan fungsionalitas dasar sistem untuk dirilis lebih awal, sehingga pengguna dapat mulai menggunakan sistem dan memberikan umpan balik yang berharga untuk iterasi selanjutnya. Ini membantu mengurangi risiko keseluruhan proyek, karena setiap iterasi memberikan kesempatan untuk mengevaluasi dan memperbaiki sistem.

Namun, Incremental Model juga memiliki tantangan tersendiri. Perencanaan yang sangat matang diperlukan untuk memastikan bahwa setiap iterasi dapat diintegrasikan dengan baik ke dalam sistem secara keseluruhan. Jika integrasi tidak dilakukan dengan benar, ini dapat menyebabkan masalah kompatibilitas antar modul, yang pada akhirnya dapat menghambat pengembangan. Selain itu, model

ini memerlukan pengelolaan yang lebih kompleks dibandingkan dengan pendekatan linear seperti Waterfall, karena pengembang harus mengelola beberapa iterasi yang mungkin memiliki kebutuhan dan jadwal yang berbeda.

Incremental Model sangat cocok untuk proyek yang memiliki tekanan untuk mengirimkan hasil dalam waktu singkat atau yang menghadapi ketidakpastian dalam kebutuhan awal. Model ini juga ideal untuk proyek yang membutuhkan iterasi terus-menerus berdasarkan umpan balik pengguna, seperti aplikasi berbasis web atau perangkat lunak yang harus berkembang sesuai dengan kebutuhan pasar. Meskipun demikian, implementasi yang sukses membutuhkan perencanaan yang hati-hati, komunikasi yang baik di antara tim, dan pengelolaan yang efektif untuk memastikan bahwa setiap iterasi memberikan nilai tambah yang berarti bagi sistem secara keseluruhan.

2.4 *Tools* yang Digunakan

2.4.1 *Hypertext Preprocessor* (PHP)

Hypertext Preprocessor (PHP) adalah bahasa pemrograman *server-side* yang digunakan secara luas untuk pengembangan web [18]. PHP awalnya dikembangkan oleh Rasmus Lerdorf pada tahun 1994 [9]. PHP dirancang untuk memungkinkan pengembang membuat halaman web yang dapat berinteraksi dengan *database*, menangani *input* dari pengguna, dan menghasilkan konten yang berbeda tergantung pada permintaan pengguna [17]. Keunggulan utama PHP terletak pada kemudahan penggunaannya dan fleksibilitasnya. PHP dapat dengan mudah disisipkan ke dalam kode HTML, sehingga memudahkan pengembang untuk mengintegrasikan kode logika pemrograman dengan elemen-elemen tampilan [16]. Ini berbeda dengan bahasa pemrograman lainnya yang membutuhkan sintaks lebih rumit untuk melakukan tugas yang sama.

Selain itu, PHP mendukung berbagai jenis *database* seperti MySQL, PostgreSQL, Oracle, dan SQLite, yang membuatnya sangat fleksibel untuk berbagai aplikasi web yang memerlukan interaksi dengan data[1]. Sebagai bahasa pemrograman *server*-side, PHP bekerja di sisi *server*, yang berarti semua

pemrosesan dilakukan di *server* dan hasilnya dikirim ke browser pengguna dalam bentuk HTML. Hal ini memberikan keuntungan dalam hal keamanan, karena kode PHP tidak terlihat oleh pengguna akhir. Meski PHP memiliki banyak keunggulan, bahasa ini juga memiliki beberapa kritik. Salah satu kritik utama adalah bahwa kode PHP yang tidak dikelola dengan baik dapat menjadi sulit untuk dibaca dan dipelihara. Selain itu, meskipun PHP telah berkembang dalam hal keamanan, bahasa ini rentan terhadap serangan jika pengembang tidak menerapkan praktik-praktik pengkodean yang aman, seperti penanganan *input* yang tidak divalidasi dengan baik, yang dapat menyebabkan serangan injeksi SQL (SQL injection) atau Cross-Site Scripting (XSS) [1].

2.4.2 MySQL *Database*

MySQL adalah sistem manajemen basis data relasional (*Relational Database Management System* atau RDBMS) yang paling populer di dunia. Dikembangkan oleh MySQL AB pada tahun 1995 dan kemudian diakuisisi oleh *Oracle Corporation* pada tahun 2010, MySQL bersifat *open-source*, yang berarti bahwa kode sumbernya dapat diakses dan dimodifikasi oleh siapa saja sesuai kebutuhan. MySQL menggunakan bahasa *Structured Query Language* (SQL) untuk mengelola, memanipulasi, dan mengambil data dalam basis data [19]. Dengan kemampuan untuk menangani data dalam jumlah besar dan skalabilitas yang baik, MySQL menjadi pilihan utama untuk berbagai aplikasi web dan bisnis, seperti situs *e-commerce*, sistem perbankan, dan aplikasi berbasis *cloud* [2]. Salah satu keunggulan utama dari MySQL adalah kecepatan dan efisiensinya dalam menangani transaksi basis data yang besar. Hal ini disebabkan oleh arsitektur penyimpanan yang mendukung berbagai mesin penyimpanan, seperti InnoDB dan MyISAM. Kemudahan penggunaan adalah faktor lain yang menjadikan MySQL sangat populer di kalangan pengembang web.

MySQL mendukung berbagai platform operasi, termasuk Windows, Linux, dan macOS, serta dapat diintegrasikan dengan berbagai bahasa pemrograman seperti PHP, Python, dan Java [3]. Selain itu, MySQL mendukung enkripsi data untuk melindungi informasi sensitif, seperti informasi kartu kredit atau data

pengguna lainnya, saat dikirimkan antara *server* dan klien. Dengan fitur-fitur keamanan ini, MySQL sangat cocok digunakan dalam aplikasi yang memerlukan perlindungan data tingkat tinggi [40]. Skalabilitas MySQL juga menjadi daya tarik utama bagi perusahaan besar. MySQL dapat menangani jutaan permintaan per detik dengan performa tinggi, yang membuatnya digunakan oleh beberapa perusahaan teknologi terbesar di dunia seperti Facebook, Twitter, dan YouTube.

Kemampuan untuk menangani data dalam skala besar dan melakukan replikasi basis data secara otomatis membuat MySQL menjadi salah satu pilihan utama untuk perusahaan yang memerlukan pengelolaan basis data besar dan kompleks. Selain itu, MySQL mendukung clustering, di mana beberapa *server* MySQL dapat bekerja secara bersamaan untuk meningkatkan keandalan dan ketersediaan basis data, bahkan dalam kondisi beban yang tinggi [41].

2.4.3 Visual Studio Code

Visual Studio Code (VS Code) adalah yang merupakan aplikasi code editor untuk membantu proses pengembangan sebuah aplikasi [42]. Dikembangkan oleh Microsoft, dirilis pertama kali pada April 2015. VS Code dirancang sebagai alat pengembangan lintas platform, yang mendukung berbagai sistem operasi seperti Windows, macOS, dan Linux [43]. Salah satu fitur utama dari VS Code adalah dukungannya terhadap ekstensi. Pengguna dapat menambahkan berbagai ekstensi untuk mendukung bahasa pemrograman atau alat-alat lain yang dibutuhkan. Ekstensi ini dapat diinstal melalui *Visual Studio Code Marketplace*, yang menawarkan ribuan ekstensi untuk bahasa seperti Python, JavaScript, C++, Java, PHP, dan banyak lainnya.

Dengan ekstensi ini, VS Code menjadi editor yang sangat fleksibel karena dapat disesuaikan dengan kebutuhan spesifik proyek pengembangan. VS Code juga dilengkapi dengan fitur IntelliSense, yang merupakan alat pelengkapan otomatis yang cerdas dan adaptif. Fitur ini tidak hanya melengkapi kode berdasarkan sintaks standar, tetapi juga memberikan informasi kontekstual tentang API, fungsi, dan variabel yang digunakan dalam proyek. Selain itu, dengan dukungan debugging bawaan, pengembang dapat melakukan debugging langsung di dalam VS Code

tanpa perlu beralih ke alat eksternal. *Debugging* ini mencakup *breakpoints, watch expressions,* dan *call stack,* yang memberikan kontrol penuh kepada pengembang dalam melacak dan memperbaiki kesalahan.

Integrasi Git yang kuat di dalam VS Code juga menjadi salah satu alasan popularitasnya. Pengguna dapat melakukan semua operasi Git dasar, seperti *commit, push, pull*, dan *merge*, langsung dari *interface* editor. Fitur ini sangat bermanfaat bagi tim pengembangan yang menggunakan sistem kontrol versi untuk berkolaborasi dalam proyek perangkat lunak. Selain itu, VS Code memiliki dukungan bawaan untuk terminal terintegrasi, memungkinkan pengembang untuk menjalankan perintah-perintah shell atau *command-line interface* (CLI) langsung di dalam editor. Ini memudahkan pengembang untuk menjalankan skrip, menguji aplikasi, atau mengelola proyek tanpa harus beralih ke terminal terpisah.

Fleksibilitas ini memberikan alur kerja yang lebih mulus dan meningkatkan produktivitas pengembang karena semua alat yang mereka butuhkan tersedia di satu tempat. Dari segi arsitektur, VS Code dibangun menggunakan teknologi web modern seperti Electron, yang memungkinkannya berjalan di berbagai platform tanpa kehilangan performa. Meskipun berbasis Electron, yang sering kali dikritik karena memakan sumber daya, VS Code terkenal karena kinerjanya yang ringan dan responsif dibandingkan dengan editor kode lain yang berbasis teknologi serupa. Dengan optimasi yang tepat, VS Code dapat menangani proyek besar dan kompleks tanpa memperlambat sistem operasi.