

BAB II

TINJAUAN PUSTAKA

2.1. Landasan Teori

2.1.1. Sistem Informasi

Sistem informasi merupakan kombinasi terintegrasi antara manusia, perangkat keras, perangkat lunak, jaringan komunikasi, serta sumber data yang digunakan untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi dalam suatu organisasi. Tujuan utama dari sistem informasi adalah untuk mendukung kegiatan operasional, pengambilan keputusan, serta pengendalian di dalam organisasi. Konsep ini menempatkan informasi sebagai sumber daya strategis yang sama pentingnya dengan modal dan tenaga kerja.

Sistem informasi modern berfungsi tidak hanya sebagai alat bantu administrasi, tetapi juga sebagai sarana untuk mencapai keunggulan kompetitif melalui transformasi digital, analitik data, dan otomatisasi proses bisnis. Sistem informasi memungkinkan organisasi untuk mempercepat respons terhadap perubahan pasar, meningkatkan efisiensi operasional, serta memperbaiki pengalaman pelanggan melalui penggunaan data yang akurat dan tepat waktu [7].

2.1.2. Tool Pengembangan Aplikasi

Dalam merancang sistem informasi manajemen limbah, pemilihan perangkat lunak (*tools*) menjadi faktor penting yang mempengaruhi efisiensi kerja, kemampuan sistem untuk berkembang (skalabilitas), serta kemudahan dalam melakukan pemeliharaan. Pada sisi *backend*, digunakan bahasa C# bersama *framework* ASP.NET Core, yang dikenal handal dalam membangun aplikasi *web enterprise* yang ringan dan bersifat modular. Sementara itu, untuk bagian *frontend*, digunakan jQuery guna

meningkatkan interaktivitas dan mendukung struktur modular dalam pengembangan berbasis JavaScript.

Pada sisi basis data, sistem ini memanfaatkan Microsoft SQL Server yang dikenal andal dalam menangani skema data relasional secara efisien dan konsisten. Sejumlah perangkat pendukung turut digunakan untuk mendukung kelancaran proses pengembangan, antara lain Visual Studio sebagai lingkungan pengembangan terintegrasi (IDE), Postman untuk pengujian antarmuka pemrograman aplikasi (API), serta Git sebagai alat kontrol versi guna memfasilitasi kolaborasi dan manajemen kode sepanjang siklus pengembangan, mulai dari tahap perancangan hingga pemeliharaan.

1. SQL Server

Microsoft SQL Server adalah sistem manajemen basis data relasional (RDBMS) yang andal dalam menangani skema data kompleks dan transaksi besar. Versi terbaru SQL Server 2022 membawa peningkatan performa, keamanan data, serta integrasi dengan Azure untuk analitik berbasis cloud [8].

2. C#

C# merupakan bahasa pemrograman modern berbasis objek yang dirancang untuk membangun berbagai aplikasi, mulai dari *desktop*, *web*, hingga *mobile*. C# 10 dan .NET 6 memperkenalkan kemampuan lintas platform, manajemen memori otomatis, serta integrasi dengan layanan *cloud*. Keunggulan C# terletak pada efisiensi kinerja, kemudahan debugging, serta kompatibilitas dengan framework .NET yang luas [9].

3. ASP.NET Core

ASP.NET Core adalah *framework open-source* yang digunakan untuk membangun aplikasi web modern dan API berbasis REST. *Framework* ini dirancang agar modular dan ringan, mendukung *dependency injection*, serta

mampu berjalan di berbagai sistem operasi. Keamanan dan skalabilitas yang tinggi menjadikannya pilihan utama untuk aplikasi *enterprise* [10].

4. JQuery

JQuery merupakan pustaka JavaScript yang dirancang untuk menyederhanakan manipulasi dokumen HTML, *event handling*, dan AJAX. bahwa meskipun banyak *framework* modern seperti React dan Vue muncul, jQuery masih relevan untuk proyek yang membutuhkan integrasi cepat dan ringan [11].

5. Trello

Trello merupakan aplikasi manajemen proyek berbasis web yang menggunakan metode Kanban untuk memudahkan kolaborasi tim. Trello efektif untuk pengorganisasian tugas, pemantauan progres proyek, serta komunikasi antar anggota tim melalui antarmuka visual yang intuitif [12].

2.1.3. Manajemen Proyek

Manajemen proyek adalah penerapan pengetahuan, keterampilan, alat, dan teknik untuk merencanakan dan mengendalikan aktivitas proyek sehingga tujuan dan persyaratan proyek dapat dipenuhi. Sebagai disiplin, manajemen proyek membentuk rangka kerja proses yang meliputi inisiasi, perencanaan, pelaksanaan, pemantauan dan pengendalian, serta penutupan; kelima kelompok proses ini sering digunakan untuk mengelola proyek ilmiah dan pengembangan sistem agar tercapai hasil yang efektif, efisien, dan terukur [13], [14].

Manajemen proyek yang efektif meningkatkan peluang keberhasilan proyek, misalnya melalui perencanaan yang rinci, pengelolaan risiko, pengaturan sumber daya, serta mekanisme komunikasi dan pengendalian mutu. Modul dan buku teks terkini menekankan pentingnya dokumentasi, perangkat bantu perencanaan seperti WBS,

jadwal, anggaran, serta proses evaluasi dan penutupan yang terdokumentasi untuk memastikan pembelajaran dan pelaporan hasil proyek [13]. Berikut tahapan tersebut:

a. Inisiasi

Mendefinisikan tujuan, ruang lingkup, pemangku kepentingan, dan keluaran yang diharapkan.

b. Perencanaan

Menyusun jadwal, biaya, sumber daya, rencana mutu, dan manajemen risiko.

c. Pelaksanaan

Menerapkan rencana termasuk pengumpulan data, pelaksanaan eksperimen/aktivitas, dan koordinasi tim.

d. Pemantauan dan Pengendalian

Membandingkan kinerja aktual dengan rencana, mengendalikan perubahan, dan melakukan pengukuran kinerja.

e. Penutupan

Menyelesaikan administrasi, menyerahkan hasil, dan mendokumentasikan pelajaran yang diperoleh [13], [14].

2.1.4. Work Breakdown Structure (WBS)

Work Breakdown Structure (WBS) adalah proses mendekomposisi proyek menjadi komponen kerja yang lebih kecil dan terstruktur untuk memudahkan pengelolaan. WBS digunakan untuk memperjelas ruang lingkup, memperkirakan biaya, dan mendefinisikan tanggung jawab [15].

2.1.5. Model Pengembangan Perangkat Lunak

Model pengembangan perangkat lunak adalah pendekatan sistematis dalam merancang dan membangun sistem informasi. Salah satu model klasik yang sering digunakan adalah model *Waterfall*, yaitu model linier-sekuensial yang menekankan

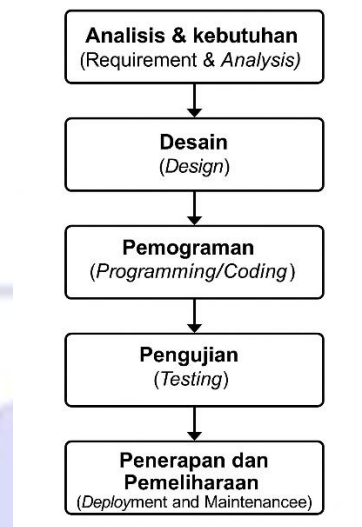
urutan fase: analisis kebutuhan, perancangan, pengkodean, pengujian, dan implementasi. Model ini cocok untuk proyek dengan kebutuhan yang stabil dan dokumentasi lengkap, namun kurang fleksibel untuk perubahan kebutuhan yang sering muncul selama pengembangan [16], [17].

Model *Waterfall* memiliki keunggulan dari segi struktur, pembagian tugas, dan *deliverables* yang jelas dan kontinu. Selain itu model ini digolongkan sebagai *predictive model* karena kejelasan di sisi tugas dan strukturnya membuat sebuah proyek dapat diprediksi pekerjaannya. Dokumentasi proyek yang dihasilkan juga cukup lengkap, detail, dan konsisten sepanjang proyek, sehingga dapat dijadikan acuan bersama [18].

Namun pendekatan ini memiliki kekurangan karena dianggap cukup kaku, tidak fleksibel terhadap penyesuaian kebutuhan aplikasi apabila prosesnya sudah berlanjut [18]. Menurut Kurniawan [18], model *Waterfall* memiliki beberapa variasi tetapi tidak jauh dari beberapa tahap utama seperti:

1. Analisis dan Kebutuhan (*Requirement and Analysis*).
2. Desain (*Design*).
3. Pemrograman (*Programming/Coding*).
4. Pengujian (*Testing*).
5. Penerapan dan Pemeliharaan (*Deployment and Maintenance*) [18].

Berikut adalah ilustrasi tahapan dalam model *Waterfall*



Sumber: [18]

2.2. Penelitian Terkait

2.2.1. Sistem Analisis Kualitas Air Limbah Industri Berbasis .NET Core dan PostgreSQL

Mengembangkan sistem untuk menganalisis kualitas air limbah industri secara online. Sistem ini menggunakan bahasa pemrograman C#, *framework* .NET Core, dan *Database* PostgreSQL. Data diperoleh melalui integrasi dengan perangkat SCADA Reliance, memungkinkan pemantauan *real-time* dan pelaporan otomatis kepada Kementerian Lingkungan Hidup dan Kehutanan (KLHK). Metodologi pengembangan

Gambar II. 1 Model *Waterfall*

yang digunakan adalah model *Waterfall* [19].

2.2.2. Sistem Manajemen Sampah TPS3R Berbasis Web dengan ReactJS dan Tailwind

Mengembangkan aplikasi web responsif untuk manajemen sampah di Tempat Pengelolaan Sampah *Reduce, Reuse, Recycle* (TPS3R) Kota Batu. Aplikasi ini dibangun menggunakan *framework* ReactJS dan Tailwind CSS, dengan metode

pengembangan *Waterfall*. Tujuan utama dari sistem ini adalah untuk mempermudah pendataan dan pelaporan jumlah sampah yang masuk dan diolah, menggantikan proses manual sebelumnya [20].

2.2.3. Implementasi Agile dalam Perancangan Sistem Pengelolaan Limbah Sampah

Menerapkan metodologi Agile dalam pengembangan sistem pengelolaan limbah sampah. Pendekatan iteratif ini memungkinkan penyesuaian berkelanjutan terhadap kebutuhan pengguna dan perubahan lingkungan. Meskipun teknologi spesifik tidak dijelaskan secara rinci, fokus penelitian ini adalah pada efektivitas metodologi Agile dalam meningkatkan kualitas dan efisiensi pengembangan sistem informasi pengelolaan limbah [21].

