

BAB II

TINJAUAN PUSTAKA

Dalam beberapa tahun terakhir, penelitian tentang pembangunan sistem manajemen proyek teknologi informasi berbasis web berkembang pesat dengan adanya *Scrum* dan arsitektur *microservices*. Pada penelitian “Sistem Informasi Manajemen Proyek PT. Samudera Perkasa Konstruksi Berbasis Web”, membahas pengembangan sistem manajemen proyek berbasis web di PT. Samudera Perkasa Konstruksil. Fokus utama sistem adalah pada pencatatan proyek, pemantauan progres, dan pengelolaan biaya. Sistem yang dikembangkan bertujuan untuk meningkatkan efisiensi dalam proses pencatatan dan pembuatan laporan, yang sebelumnya dilakukan secara manual [7].

Dalam penelitian yang berjudul “Pengembangan Sistem Aplikasi Manajemen Proyek Berbasis Web (Studi Kasus: PT. Swadaya Graha)”, sistem yang dikembangkan difokuskan untuk membantu perencanaan, pengawasan, dan pelaporan proyek secara menyeluruh. Penelitian ini menyoroti pentingnya sistem terintegrasi yang mampu mengelola berbagai aspek proyek, mulai dari penjadwalan, penugasan tugas, hingga evaluasi progres kerja. Hasil pengujian menunjukkan bahwa sistem yang dibangun tidak hanya memberikan kemudahan dalam penggunaan (*usability*), tetapi juga mampu meningkatkan transparansi dan efisiensi dalam pelaksanaan proyek. Sistem ini dirancang dengan antarmuka yang *user-friendly* serta alur kerja yang sesuai dengan kebutuhan operasional di lapangan, sehingga memudahkan pengguna dalam memahami dan mengoperasikan fitur-fitur yang tersedia [8].

Dalam penelitian yang berjudul “Pengembangan Aplikasi dan Website Manajemen Proyek PT Santai Berkualitas Syberindo Menggunakan Metode *Agile*”, menunjukkan bahwa sistem manajemen proyek berbasis web yang dibangun menggunakan metode *Agile* dapat membantu mengatasi permasalahan manajemen proyek yang sedang berlangsung. Metode *Agile* dipilih karena kemampuannya untuk beradaptasi dengan perubahan kebutuhan pengguna serta memungkinkan pengembangan berulang sesuai dengan situasi yang muncul selama proyek berjalan. Laravel (*backend*) yang digunakan dalam pengembangan website ini

memungkinkan manajer proyek untuk mengelola data proyek secara. Dengan implementasi sistem ini, PT. Santai Berkualitas Syberindo diharapkan dapat meningkatkan efisiensi dan kualitas manajemen proyek mereka [9].

Pada penelitian “Pengembangan Sistem Manajemen Proyek Perangkat Lunak (Studi Kasus: CV. Karya Studio Teknologi Digital)”, menyebutkan bahwa salah satu masalah utama yang dihadapi adalah kesulitan manajer proyek dalam memantau progres tugas yang dikerjakan oleh *programmer*. Hal ini disebabkan oleh tidak adanya sistem yang menghimpun laporan progres secara terstruktur, sehingga semua informasi menumpuk di media grup percakapan. Untuk mengatasi masalah tersebut, dikembangkan sistem manajemen proyek perangkat lunak berbasis web. Penelitian menunjukkan bahwa sistem ini dapat membantu memantau dan mengelola proyek secara lebih terstruktur dan transparan antara tim pengembang dan klien [10].

Pada penelitian “Sistem Informasi Manajemen Proyek Menggunakan Teknologi MERN *stack* pada PT. Wahanakarsa Swandiri”, menghasilkan sebuah aplikasi berbasis web untuk memanajemen proyek di PT. Wahanakarsa Swandiri. Aplikasi ini membantu perusahaan dalam pengelolaan proyek, terutama dalam memantau progres dan mencatat pengeluaran. Sistem ini dilengkapi dengan beberapa level pengguna, yaitu Administrator, *Manager*, *Team Leader*, dan *Field Worker*, yang masing-masing memiliki peran yang berbeda dalam manajemen proyek. Aplikasi ini memberikan manfaat dalam pencatatan tugas harian, progres proyek, pengeluaran keuangan, serta sebagai alat informasi digital untuk distribusi tugas proyek kepada karyawan [11].

Pada penelitian “*Design and Development of Software Project Management System using Scrum*”, menyimpulkan bahwa sistem manajemen proyek berbasis metode *Scrum* yang dikembangkan menggunakan Laravel mampu membantu organisasi dalam mengelola proyek dengan efisien. Analisis deskriptif terhadap tanggapan pengguna menunjukkan bahwa respon pengguna terhadap sistem manajemen proyek ini positif, sehingga sistem ini dapat meningkatkan efektivitas dalam pengelolaan proyek perangkat lunak [12].

Pada penelitian “Penerapan Arsitektur *Microservice* Pada Sistem Tata Kelola Matakuliah Proyek Politeknik Pos Indonesia”, mengkaji Sistem Tata Kelola Proyek di Politeknik Pos Indonesia pada program studi D4 Teknik Informatika. Sistem informasi yang mengelola proyek tersebut telah diuji dan beroperasi dengan baik dari awal hingga akhir kegiatan proyek. Namun, terdapat masalah dalam proses pengembangan fitur dan penanganan kerusakan sistem, terutama karena lamanya waktu yang dibutuhkan akibat banyaknya penyesuaian di berbagai bagian sistem, baik *frontend* maupun *backend*. Tantangan ini menjadi lebih signifikan ketika pengembang baru diperlukan untuk melakukan pengembangan fitur, karena sistem yang ada memerlukan penyesuaian yang kompleks seiring berkembangnya aplikasi. Untuk mengatasi hal ini, rencana penerapan arsitektur *microservices* diajukan. Dengan arsitektur ini, sistem informasi diharapkan lebih mudah dipelajari oleh pengembang baru, khususnya dengan penggunaan bahasa pemrograman JavaScript[13].

Pada penelitian berjudul “*Application of Microservice Architecture in Cloud Environment Project Development*”, membahas tantangan yang dihadapi dalam pengembangan dan pemeliharaan sistem bisnis yang semakin kompleks di era informasi saat ini. Untuk mengatasi masalah tersebut, diusulkan sebuah *platform* pengembangan aplikasi yang terintegrasi berdasarkan arsitektur *microservices*. Dibandingkan dengan arsitektur monolitik tradisional, arsitektur *microservices* memungkinkan pemisahan sistem aplikasi yang besar dan kompleks menjadi modul-modul layanan yang dapat dikembangkan, diuji, dioperasikan, dan diperbarui secara independen. Ini memberikan metode yang lebih efektif dalam menangani skala dan kompleksitas aplikasi di perusahaan-perusahaan besar, terutama perusahaan berbasis internet. Studi kasus tentang pengembangan sistem *cloud platform* dijelaskan untuk menunjukkan bagaimana arsitektur ini diterapkan dalam proyek nyata, serta keunggulan yang dimiliki arsitektur *microservices* dibandingkan arsitektur monolitik. Hasil dari penelitian ini menyimpulkan bahwa arsitektur *microservices* memberikan panduan yang signifikan dalam menyelesaikan tantangan yang sering muncul dalam pengembangan aplikasi di tingkat perusahaan [14].

Pada penelitian dengan judul “*Microservices Architecture for Cloud Computing*”, membahas tentang arsitektur *Microservices* yang dirancang untuk memenuhi kebutuhan aplikasi modern dengan ketersediaan tinggi, skalabilitas, dan kemampuan dieksekusi dengan mudah di *platform cloud*. *Microservices* memungkinkan pengembangan aplikasi yang terdiri dari beberapa layanan kecil yang dapat dikembangkan secara independen, dengan fokus pada bagian spesifik dari aplikasi. Arsitektur ini memberikan fleksibilitas pengembang dalam mengembangkan aplikasi[15].

Pada buku “*Containerize your Apps with Docker and Kubernetes*” karya Dr. Gabriel N. Schenker, membahas tentang bagaimana cara menerapkan aplikasi berbasis *container* menggunakan *Azure Cloud Services*. Buku elektronik ini membahas tentang cara bekerja dengan *container* berbasis *image* yang sudah ada. Buku ini menjelaskan beberapa operasi dasar seperti menjalankan, menghentikan, memulai kembali, dan menghapus *container* [16].

Pada jurnal dengan judul “Implementasi Arsitektur *Microservice* pada Aplikasi Web Pengajaran Agama Islam *Home Pesantren*”, membahas tentang pengembangan aplikasi web pengajaran agama Islam "Home Pesantren" yang sebelumnya menggunakan arsitektur monolitik. Arsitektur tersebut menempatkan seluruh komponen aplikasi dalam satu proyek yang sama, namun memiliki kelemahan dalam hal skalabilitas. Ketika aplikasi membutuhkan peningkatan skala, seluruh sistem harus ditingkatkan secara keseluruhan, yang menyebabkan ketergantungan yang tinggi antar layanan. Untuk mengatasi masalah ini, rencana penerapan arsitektur *microservice* dilakukan agar aplikasi menjadi lebih modular dan lebih mudah untuk diskalakan[17]. Data mengenai perbandingan antara penelitian sebelumnya dengan sistem yang akan dibangun dapat dilihat pada Tabel 2.1 sebagai berikut.

Tabel II.1 Perbandingan Penelitian Sebelumnya

No	Penelitian Sebelumnya	Teknologi	Fokus Utama Penelitian Sebelumnya	Sistem yang Dibangun Dalam Penelitian Ini
----	--------------------------	-----------	--	---

1	Sistem Informasi Manajemen Proyek PT. Samudera Perkasa Konstruksi [6]	Web (PHP/MySQL)	Pengelolaan proyek dan laporan manual	Menggunakan arsitektur <i>microservices</i> dan <i>cloud deployment</i> (Azure)
2	Pengembangan Aplikasi Proyek PT Santai Berkualitas Syberindo [8]	Laravel + <i>Agile</i>	Manajemen proyek dengan metode <i>Agile</i>	Fokus pada peran <i>Scrum</i> lengkap (<i>product owner</i> , <i>scrum master</i> , <i>developer</i>) dan <i>board</i> interaktif
3	Sistem Manajemen Proyek CV Karya Studio Teknologi Digital [9]	Web monolitik	Pemantauan progres tugas	Sistem modular, tiap layanan (<i>auth</i> , <i>backlog</i> , <i>sprint</i>) berjalan terpisah
4	<i>Microservices</i> untuk Tata Kelola Matakuliah Politeknik Pos [12]	<i>Microservices</i> (JS)	Peralihan arsitektur monolitik ke <i>microservice</i>	Sudah diimplementasikan penuh dengan komunikasi antar layanan via gRPC & REST API
5	Sistem Manajemen Proyek PT. Wahanakarsa Swandiri [10]	<i>MERN Stack</i>	Pencatatan tugas, progres, dan pengeluaran	Lebih ringan, tidak membebani <i>user</i> dengan level akses rumit & fitur tak perlu

Berdasarkan Tabel 2.1, penelitian ini memiliki nilai orisinalitas yang kuat dibandingkan dengan penelitian-penelitian sebelumnya karena menggabungkan pendekatan *Scrum* yang utuh, arsitektur *microservices*, dan *deployment* berbasis *Azure Cloud* ke dalam satu sistem manajemen proyek yang benar-benar terimplementasi. Jika sebagian besar penelitian terdahulu berfokus pada pencatatan proyek secara umum atau pengelolaan proyek tradisional, sistem yang dibangun dalam skripsi ini secara spesifik menyesuaikan fitur-fitur yang sesuai dengan

praktik *Scrum* modern, termasuk fitur seperti manajemen *sprint*, *backlog*, *product goal*, hingga tampilan *board* dan *timeline*.

Selain itu, sistem ini juga diuji tingkat ketergantungannya, seperti saat layanan *email* mati namun sistem autentikasi tetap berjalan—hal yang jarang dibahas atau diimplementasikan secara nyata dalam penelitian lain. Dengan kata lain, sistem dibangun agar mudah digunakan, modular, dan memiliki ketergantungan antar layanan yang rendah sesuai dengan tujuan utama penelitian.

