

BAB II

LANDASAN TEORI

2.1 Landasan Teori

2.1.1 PT Imbang Tata Alam

PT Imbang Tata Alam adalah sebuah perusahaan yang bergerak di bidang minyak dan gas bumi. Perusahaan ini memiliki hak kerja sama untuk mengelola wilayah Kontrak Kerja Sama (KKS) Malacca Strait yang berada di Provinsi Riau. Kegiatan utama perusahaan ini adalah mencari dan memproduksi energi dari sumber daya alam yang ada di daerah tersebut.

Sebagai bagian dari sektor energi, PT Imbang Tata Alam memiliki peran penting dalam membantu memenuhi kebutuhan energi di Indonesia. Perusahaan ini menjalankan proses eksplorasi dan produksi dengan tetap memperhatikan standar keselamatan kerja serta menjaga kelestarian lingkungan sekitar.

Dalam menjalankan operasionalnya, PT Imbang Tata Alam juga bekerja sama dengan berbagai pihak untuk memastikan kegiatan berjalan dengan efisien dan bertanggung jawab. Perusahaan terus berusaha memberikan kontribusi positif, baik bagi negara maupun masyarakat di sekitar wilayah operasionalnya. Salah satu dokumentasi dari PT. Imbang Tata Alam dapat dilihat pada Gambar 2.1 PT. Imbang Tata Alam di bawah berikut.



Gambar 2.1 PT. Imbang Tata Alam

2.1.2 Sistem Manajemen Aset

Sistem manajemen aset adalah sebuah sistem yang dirancang untuk membantu perusahaan mencatat, memantau, dan mengelola aset yang dimiliki secara lebih teratur dan efisien. Sistem ini mencakup berbagai proses mulai dari pencatatan saat aset masuk, pemeliharaan selama aset digunakan, hingga penghapusan aset jika sudah tidak layak pakai. Dengan sistem yang baik, perusahaan bisa menghindari kehilangan aset dan memperpanjang masa pakai barang.

Selain itu, sistem manajemen aset juga memudahkan perusahaan dalam membuat laporan terkait kondisi dan nilai aset yang dimiliki. Laporan ini penting untuk pengambilan keputusan, seperti kapan harus melakukan perawatan atau penggantian aset. Penggunaan teknologi modern seperti sistem berbasis web membuat proses ini bisa dilakukan dengan cepat dan dapat diakses kapan saja oleh pihak yang berwenang.

Menurut penelitian Mardivta dkk. (2022), penerapan sistem informasi manajemen aset berbasis web di PT. Bukit Asam Tbk. berhasil meningkatkan efisiensi pengelolaan aset dan mengurangi kesalahan dalam pencatatan. Sistem ini memungkinkan proses monitoring aset berjalan lebih lancar dan data yang dihasilkan lebih akurat.

2.1.3 Backend System

Backend adalah bagian dari aplikasi yang bekerja di belakang layar dan tidak langsung terlihat oleh pengguna, namun memiliki peran yang sangat penting. Tugas backend adalah mengatur data, melakukan perhitungan, dan menjalankan proses di server, misalnya saat pengguna mendaftar akun atau melihat informasi. Backend mengambil data dari database, memprosesnya, lalu mengirim hasilnya kembali ke tampilan pengguna.

Backend juga berfungsi sebagai penghubung antara aplikasi dan *database* agar data bisa disimpan dan diambil dengan baik. Agar cepat, aman, dan mudah dirawat, *backend* biasanya dibuat menggunakan *REST API*. *REST API* membantu aplikasi dan *server* bertukar data dengan format yang ringan seperti *JSON*, sehingga komunikasi antar sistem menjadi mudah dan efisien.

Menurut Herdiyatmoko (2022), penggunaan *REST API* dalam desain *backend* sangat membantu pengembangan aplikasi karena lebih efisien, mudah dikembangkan dan dirawat, serta dapat digunakan oleh berbagai jenis *platform*. Artinya, satu *backend* bisa digunakan oleh banyak aplikasi yang berbeda tanpa perlu dibuat ulang dari awal.

2.1.4 Scrum

Scrum adalah salah satu metode pengembangan perangkat lunak yang bersifat iteratif dan incremental. Metode ini memecah pekerjaan besar menjadi bagian-bagian kecil yang dikerjakan secara bertahap dalam periode waktu tertentu yang disebut sprint.

Salah satu hal yang sangat ditekankan dalam metode pengembangan ini adalah komunikasi yang rutin antar anggota tim melalui pertemuan singkat harian, sehingga progres dan hambatan bisa langsung diketahui dan diselesaikan. Metode ini juga fleksibel, sehingga memungkinkan perubahan arah pengembangan jika diperlukan agar hasil akhir sesuai dengan kebutuhan pengguna. Berikut proses lengkap Scrum dari awal sampai akhir.

1. *Product Backlog*

Product backlog adalah daftar semua fitur, tugas, atau kebutuhan yang harus dikerjakan dalam proyek. *Product Owner* bertanggung jawab membuat dan mengatur daftar ini berdasarkan prioritas kebutuhan bisnis dan pengguna. Daftar ini terus diperbarui selama proyek berjalan.

2. *Planning*

Pada tahap ini, tim Scrum memilih item-item dari *product backlog* yang akan dikerjakan pada sprint berikutnya. Tim mendiskusikan tujuan sprint dan menetapkan pekerjaan apa yang realistis untuk diselesaikan selama sprint berlangsung.

3. *Sprint*

Sprint adalah waktu kerja aktif yang biasanya berlangsung antara 1 sampai 4 minggu. Dalam periode ini, tim fokus mengerjakan tugas yang sudah direncanakan tanpa gangguan dari luar agar hasil kerja maksimal.

4. *Daily Scrum*

Setiap hari, tim melakukan pertemuan singkat sekitar 15 menit. Dalam pertemuan ini, anggota tim melaporkan apa yang sudah dikerjakan, rencana kerja hari ini, dan kendala yang dihadapi. Pertemuan ini membantu menjaga komunikasi tetap lancar dan masalah cepat diatasi.

5. *Sprint Review*

Setelah *sprint* selesai, tim mempresentasikan hasil kerja kepada pemangku kepentingan atau pengguna. Tujuannya adalah untuk mendapatkan masukan dan memastikan hasil yang dibuat sesuai dengan kebutuhan dan harapan.

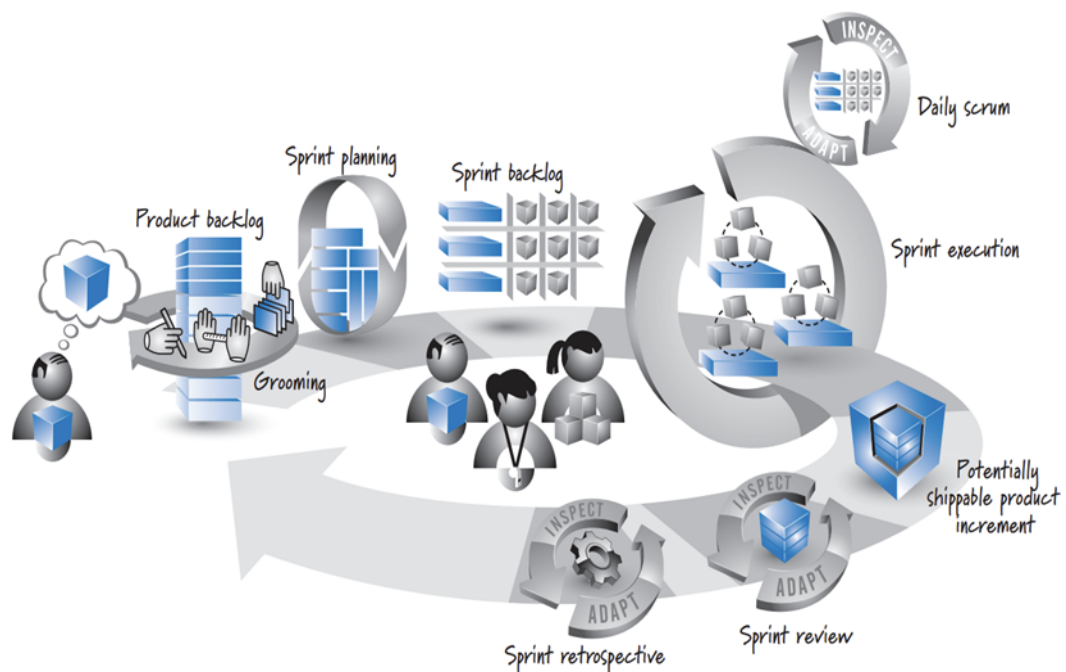
6. *Sprint Retrospective*

Ini adalah pertemuan evaluasi internal tim untuk membahas apa yang sudah berjalan baik dan apa yang perlu diperbaiki. Tujuannya agar proses kerja terus membaik di *sprint* berikutnya.

7. Perulangan *Sprint*

Setelah satu siklus *sprint* selesai, tim kembali ke tahap *sprint planning* dan mengulangi seluruh proses ini sampai semua fitur selesai dan produk akhir tercapai.

Penelitian oleh Wicaksono dkk. (2023) menunjukkan bahwa penerapan metode Scrum dalam pengembangan sistem informasi manajemen aset di Telkom University berhasil meningkatkan efisiensi proses pengembangan. Ilustrasi dari *Scrum* dapat dilihat pada Gambar 2.2 Metode *Scrum* di bawah berikut.



Gambar 2.2 Metode Scrum

Sumber: *Essential Scrum: A Practical Guide to the Most Popular Agile Process* (Kenny Rubin, 2012)

2.1.5 Integration Testing

Integration Testing adalah tahap pengujian perangkat lunak yang dilakukan setelah unit testing, dengan tujuan untuk memastikan bahwa setiap modul atau komponen yang telah diuji secara terpisah dapat bekerja sama dengan baik saat digabungkan. Pengujian ini berfokus pada interaksi antar modul, aliran data, dan komunikasi antar komponen dalam sistem untuk mendeteksi adanya kesalahan integrasi.

Metode ini sangat penting karena meskipun setiap modul telah berfungsi dengan baik secara individual, kesalahan dapat muncul ketika modul-modul tersebut saling berinteraksi. Contohnya, pada sistem informasi manajemen aset, pengujian integrasi dapat dilakukan antara modul login dengan modul pengelolaan data aset untuk memastikan proses autentikasi dan akses data berjalan konsisten.

Menurut penelitian yang dilakukan oleh Pratama dkk. (2022), penerapan *integration testing* pada sistem informasi berbasis web mampu mendeteksi kesalahan interaksi antar modul sejak tahap awal pengujian. Pendekatan ini juga memastikan bahwa proses pertukaran data antar komponen sistem berjalan

konsisten dan sesuai dengan fungsinya. Dengan demikian, *integration testing* menjadi tahapan krusial dalam menjamin kestabilan serta keandalan sistem sebelum dilanjutkan ke tahap pengujian berikutnya.

2.1.6 *Performance Testing*

Performance Testing adalah proses pengujian yang dilakukan untuk mengukur sejauh mana sistem mampu beroperasi dengan baik di bawah kondisi tertentu, seperti beban pengguna yang tinggi, volume data besar, atau waktu respons yang cepat. Tujuannya adalah untuk memastikan bahwa sistem memiliki kinerja yang stabil, responsif, dan efisien dalam menangani berbagai situasi penggunaan.

Dalam konteks sistem informasi manajemen aset, performance testing dapat mencakup pengujian waktu akses halaman, kecepatan pemrosesan data aset, serta kestabilan sistem saat banyak pengguna melakukan aktivitas secara bersamaan. Pengujian ini membantu mengidentifikasi potensi bottleneck dan memastikan sistem tetap berfungsi optimal dalam lingkungan operasional sebenarnya.

Menurut penelitian yang dilakukan oleh Ramadhan dkk. (2022), penerapan *performance testing* pada sistem informasi manajemen terbukti dapat meningkatkan efisiensi pemrosesan data serta mempercepat waktu respon sistem terhadap interaksi pengguna. Pengujian ini memastikan sistem mampu menangani beban kerja yang tinggi tanpa menurunkan kualitas layanan. Dengan demikian, *performance testing* menjadi langkah penting dalam menjaga stabilitas kinerja dan pengalaman pengguna terhadap sistem secara menyeluruh.

2.1.7 *Security Testing*

Security Testing adalah proses pengujian yang dilakukan untuk memastikan bahwa sistem memiliki tingkat keamanan yang memadai dan mampu melindungi data dari ancaman seperti akses tidak sah, kebocoran informasi, maupun serangan siber. Tujuan utama pengujian ini adalah untuk menemukan celah keamanan serta memastikan mekanisme autentikasi, otorisasi, dan enkripsi data berjalan benar.

Dalam konteks sistem informasi manajemen aset, security testing dapat mencakup pengujian modul autentikasi pengguna, pengelolaan hak akses (role-based access control), enkripsi data aset dan log audit, serta pengujian terhadap serangan seperti injeksi (SQL Injection), Cross-Site Scripting (XSS), atau pengambilan alih sesi. Pengujian ini penting agar data perusahaan tetap aman, sistem stabil, dan tidak terjadi pelanggaran keamanan yang merugikan.

Menurut penelitian oleh Nasution dkk. (2023), banyak sistem informasi manajemen di Indonesia masih memiliki celah keamanan yang dapat dimanfaatkan untuk serangan siber, terutama pada bagian autentikasi dan perlindungan data sensitif. Oleh karena itu, penerapan security testing menjadi langkah penting untuk mendeteksi dan menutup kerentanan tersebut sejak tahap pengembangan sistem, sehingga integritas dan kerahasiaan data dapat terjaga.

2.1.8 *Application Programming Interface (API)*

API adalah antarmuka yang memungkinkan dua sistem atau aplikasi berbeda untuk saling berkomunikasi dan bertukar data dengan mudah. Dalam pengembangan *backend*, *API* berfungsi sebagai penghubung antara *client* dan server. Jadi ketika pengguna melakukan suatu aksi di aplikasi, seperti menekan tombol, permintaan itu akan dikirim ke server melalui *API*. Server kemudian memproses permintaan tersebut dan mengirimkan data atau hasilnya kembali ke *client* lewat *API* juga. Dengan cara ini, komunikasi antara frontend dan backend bisa berjalan dengan lancar.

Salah satu contoh yang umum adalah penggunaan layanan peta seperti *Google Maps* dalam aplikasi ojek online. Ketika pengguna ingin menentukan lokasi penjemputan, aplikasi tidak membuat sistem peta sendiri. Sebaliknya, aplikasi akan menggunakan *API* dari *Google Maps* untuk menampilkan peta dan informasi lokasi. Jadi, tanpa harus membangun ulang sistem dari nol, aplikasi bisa langsung menggunakan layanan yang sudah ada melalui *API*.

Menurut (Priyatna, 2020), *API* merupakan titik interaksi antara aplikasi yang digunakan untuk bertukar data dan layanan dalam sistem terintegrasi. Oleh karena itu, *API* memudahkan pertukaran informasi antar aplikasi sehingga proses pengolahan data menjadi lebih efisien dan sistem dapat berjalan secara terintegrasi.

2.1.9 *Laravel*

Laravel adalah *framework PHP* yang sangat populer dan banyak digunakan untuk membangun aplikasi web. *Framework* ini dirancang untuk memudahkan pengembang dalam membuat aplikasi dengan menyediakan berbagai fitur siap pakai. Misalnya, *Laravel* memiliki sistem *routing* yang membantu mengatur alur halaman web, pengelolaan *database* yang memudahkan penyimpanan dan pengambilan data, serta fitur keamanan untuk melindungi aplikasi dari serangan. Dengan adanya fitur-fitur ini, pengembang tidak perlu membuat semuanya dari awal, sehingga waktu pengembangan bisa lebih cepat dan efisien.

Salah satu keunggulan utama *Laravel* adalah penerapan desain *MVC* (*Model-View-Controller*) yang membagi aplikasi menjadi tiga bagian utama agar lebih terstruktur. Bagian Model bertugas mengelola data dan aturan bisnis, View mengatur tampilan yang dilihat oleh pengguna, dan *Controller* berfungsi sebagai penghubung yang mengatur alur data antara Model dan View. Dengan pembagian tugas yang jelas ini, kode menjadi lebih mudah dipahami, dirawat, dan dikembangkan, terutama saat aplikasi mulai bertambah besar dan rumit.

Menurut penelitian oleh Zulkhaidi dkk. (2019), implementasi *Laravel* pada sistem informasi penjualan produk elektronik berbasis web terbukti mempercepat proses pengembangan dan memudahkan pemeliharaan kode. *Laravel* menyediakan plugin dan struktur *MVC* yang memudahkan perancangan *database* secara terpisah dari tampilan, serta fitur *routing*, autentikasi, dan manajemen data, sehingga pengembangan menjadi lebih cepat dan aman.

2.1.10 *Middleware*

Middleware adalah perangkat lunak perantara yang berfungsi menghubungkan berbagai komponen dalam sistem *backend*. *Middleware* sering digunakan untuk menangani autentikasi, pencatatan log, dan penanganan kesalahan. *Framework backend* modern seperti Express.js, *Laravel*, dan Django umumnya memanfaatkan *middleware* untuk mengelola alur permintaan dan respons secara efisien.

Menurut Herdiyatomoko (2022), *middleware* dapat menyisipkan fungsi tambahan di antara proses permintaan dan respons, sehingga mempermudah

pengelolaan alur data dalam aplikasi. Dengan demikian, *middleware* membantu menjaga kode tetap terstruktur dan memudahkan pemeliharaan serta pengembangan sistem *backend*.

2.1.11 Otentikasi dan Otorisasi

Otentikasi dan otorisasi merupakan dua komponen penting dalam menjaga keamanan sistem backend. Otentikasi berfungsi untuk memverifikasi identitas pengguna, memastikan bahwa hanya pengguna yang sah yang dapat mengakses sistem. Setelah identitas pengguna terverifikasi, otorisasi menentukan hak akses pengguna tersebut terhadap sumber daya sistem, memastikan mereka hanya dapat mengakses data atau fungsi yang sesuai dengan izin yang diberikan.

Menurut penelitian oleh Nashikhuddin dkk. (2023) , implementasi JSON Web Token (JWT) dalam API REST efektif dalam menangani proses otentikasi dan otorisasi pengguna. JWT membantu memastikan bahwa hanya pengguna yang telah terverifikasi yang dapat mengakses data sensitif, sehingga meningkatkan keamanan aplikasi.

2.1.12 Postman

Postman adalah sebuah alat (tool) yang digunakan untuk melakukan pengujian terhadap *Application Programming Interface (API)* pada pengembangan perangkat lunak. *Postman* memungkinkan pengembang untuk mengirim permintaan (*request*) seperti *GET*, *POST*, *PUT*, dan *DELETE* ke server, serta menerima tanggapan (*response*) yang dikirim oleh sistem. Dengan cara ini, pengembang dapat memastikan bahwa *API* bekerja sesuai dengan fungsinya sebelum diintegrasikan ke antarmuka pengguna.

Dalam pengembangan sistem informasi manajemen aset, *Postman* berperan penting untuk menguji komunikasi antara *frontend* dan *backend*, seperti pengambilan data aset, proses autentikasi, hingga pembaruan data. Melalui *Postman*, pengembang dapat memverifikasi bahwa setiap *endpoint API* memberikan hasil sesuai harapan, baik dari sisi data maupun status respons, sehingga kesalahan dalam pertukaran data dapat diminimalkan.

Menurut penelitian oleh Sulistiyowati dan Rahmawati (2022), penggunaan *Postman* membantu meningkatkan efisiensi pengujian *API* karena mampu

menampilkan hasil pengujian secara *real-time* dan mendukung otomatisasi skenario uji. Dengan demikian, *Postman* menjadi salah satu alat yang efektif untuk memastikan kestabilan dan keandalan komunikasi antar komponen sistem.

2.2 Penelitian Terdahulu

Penelitian oleh Putri dkk. (2023) mengembangkan sistem informasi inventaris berbasis *web* menggunakan *framework CodeIgniter*. Tujuan utama penelitian ini adalah untuk membantu pengelolaan barang di Pusat Pendidikan dan Pelatihan Pajak (PPPP). Hasilnya, sistem *backend* yang dibuat berhasil mempercepat proses pengolahan data inventaris dan menyajikan informasi dengan lebih efisien. Teknologi yang digunakan antara lain *PHP* dan *MySQL*.

Sedangkan penelitian oleh Setiawan dkk. (2023) yang membahas tentang implementasi *Scrum model* dalam pengembangan aplikasi *e-commerce* pada bidang jasa pembangunan rumah menggunakan metode *Scrum* untuk mengembangkan aplikasi berbasis *web*. Tujuan utama penelitian ini adalah menciptakan sistem yang mampu meningkatkan efisiensi transaksi antara pelanggan dan penyedia jasa. Hasilnya, penerapan *Scrum* membuat proses pengembangan lebih terstruktur, meningkatkan kolaborasi tim, dan mempercepat penyelesaian fitur sesuai kebutuhan pengguna.

Kemudian, penelitian yang dilakukan oleh Ani dkk. (2023) berfokus pada penerapan *framework Yii* dalam membangun sistem inventaris di SMK Negeri 2 Prabumulih. Tujuan penelitian ini adalah untuk menciptakan *backend* yang mampu mendukung pengelolaan inventaris barang sekolah dengan lebih efisien. Hasil dari penelitian ini menunjukkan bahwa penggunaan *Yii Framework* pada *backend* berhasil meningkatkan efisiensi dalam pengelolaan data inventaris, sehingga proses pencatatan dan pemantauan barang menjadi lebih cepat dan terorganisir.

Terakhir, Hartati (2024) juga melakukan penelitian serupa dengan membangun sistem informasi inventaris di PT Buana Eltra. Sistem *backend* dikembangkan menggunakan *PHP* dan *MySQL*. Tujuan utama penelitian ini adalah untuk membantu para karyawan dalam mendata dan melacak barang-barang inventaris. Hasilnya, sistem berhasil meningkatkan efisiensi dan akurasi dalam pencatatan barang.

Berdasarkan kajian terhadap beberapa penelitian terdahulu, diketahui bahwa penelitian-penelitian tersebut telah membahas topik yang relevan mengenai pengelolaan inventaris dan penggunaan metode *Scrum*. Namun, penelitian sebelumnya masih memiliki beberapa keterbatasan. Sebagian besar penelitian terdahulu masih menggunakan metode pengembangan sistem secara konvensional, belum menggunakan arsitektur *web service* berbasis REST API yang memisahkan *backend* dan *frontend*, serta belum dilengkapi dengan pengaturan hak akses berbasis peran.

Penelitian ini memiliki perbedaan dengan penelitian terdahulu karena berfokus pada pengembangan *backend* untuk mendukung pengelolaan aset operasional di PT Imbang Tata Alam. Sistem dikembangkan menggunakan *web service* REST API dengan *framework* Laravel dan basis data MySQL serta menerapkan metode *Scrum*. Pengembangan *backend* mencakup pengelolaan data, proses pengajuan aset, pengaturan hak akses, dan penyediaan API untuk mendukung kebutuhan pengguna. Perbandingan penelitian terdahulu dengan penelitian ini dapat dilihat pada Tabel 2.1 Perbandingan Penelitian Terdahulu.

Tabel 2.1 Perbandingan Penelitian Terdahulu

Penulis	Judul	Metodologi	Studi Kasus	Hasil
Nabillah A. Putri dkk. (2023)	Sistem Informasi Inventaris Barang menggunakan Code-igniter pada Pusat Pendidikan dan Pelatihan Pajak (PPPP)	RAD (Rapid Application Development)	Pusat Pendidikan dan Pelatihan Pajak (PPPP)	Sistem backend berbasis CodeIgniter dan MySQL mempermudah pengolahan data inventaris serta mempercepat penyampaian informasi

Penulis	Judul	Metodologi	Studi Kasus	Hasil
				mengenai persediaan barang.
Setiawan dkk (2023)	Implementasi Scrum model dalam pengembangan aplikasi e-commerce pada bidang jasa pembangunan rumah	Scrum	Aplikasi e-commerce jasa pembangunan rumah	Framework Scrum membuat alur kerja lebih jelas, meningkatkan kolaborasi tim, dan mempermudah update sistem
Astriani (2023)	Implementasi Framework Yii dalam Membangun Sistem Inventaris pada SMK Negeri 2 Prabumulih	User Centered Design	SMK Negeri 2 Prabumulih	Backend Yii Framework meningkatkan efisiensi pengelolaan inventaris barang sekolah.
Hartati (2024)	Membangun Sistem Informasi Inventaris Barang pada PT. Buana MySQL	Waterfall	PT. Buana Eltra	Backend PHP dan MySQL mempermudah pendataan inventaris barang secara efisien