

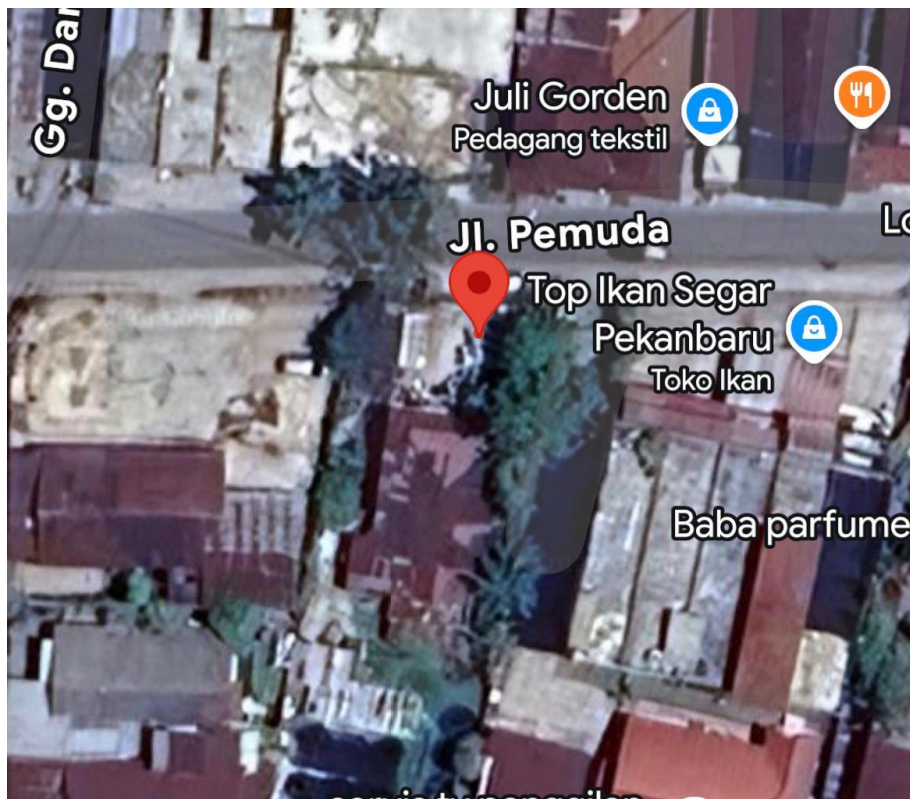
BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Latar Belakang PT. Muda Raya Gemilang

PT. Muda Raya Gemilang merupakan perusahaan swasta yang berdiri tahun 2024. Sebelum berdirinya PT ini, Bapak Un Hong sudah lama bergerak di bidang konstruksi selama 20 Tahun. Perusahaan ini memiliki kantor yang berlokasi di Jl Pemuda No 10, Kec. Payung Sekaki, Kota Pekanbaru, Riau. Sejak awal berdirinya, PT Muda Raya Gemilang berfokus pada proyek konstruksi rumah dan lain-lainnya di wilayah pekanbaru dan sekitarnya. Pada saat ini, PT Muda Raya Gemilang sudah memiliki 60 karyawan dan memiliki portofolio yang tersebar di berbagai kabupaten yang sudah membangun ruko sudirman, kubang raya, singgalang, arengka dan juga ikut membangun rebung residence dan cakra residence. Pada gambar 2.1 titik lokasi berwarna merah menunjukkan lokasi dari PT Muda Raya Gemilang.



Gambar 2.1 Lokasi PT Muda Raya Gemilang

2.1.2 Sistem Informasi

Menurut Ramadhan, 2017 sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dalam kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat yang tertentu. Untuk memperoleh informasi yang berguna, tindakan yang pertama adalah mengumpulkan data, kemudian mengolahnya sehingga menjadi informasi. Dari data-data tersebut informasi yang didapatkan lebih terarah dan penting karena telah dilalui berbagai tahap dalam pengolahannya diantaranya yaitu pengumpulan data, data apa yang terkumpul dan menemukan informasi yang diperlukan.

Pengolahan data saat ini masih menggunakan pencatatan secara manual di atas kertas yang sering kali terjadi kesalahan dan keterlambatan dalam proses pembuatan laporan dan juga beresiko besar dalam keamanan data sehingga dinilai kurang efektif dan efisien waktu (Maydianto & Ridho, 2021). Dari permasalahan ditemukan, maka perlu dibuat satu sistem informasi yang dapat membantu pemilik maupun pengguna untuk menyelesaikan masalah dan menjadi lebih efektif.

2.1.3 Procurement

Menurut (Ibrahim & Phahlevy, 2024), *procurement* merupakan kegiatan yang meliputi proses perencanaan, pemilihan vendor, hingga pembelian barang. Tujuan dari *procurement* sendiri untuk memastikan barang yang diperoleh dengan biaya dan kualitas yang bagus. *Procurement* sendiri selalu menjadi bagian dari perusahaan karena berkaitan dengan operasional dan layanan barang. Oleh karena itu, *procurement* yang sistematis dapat memberikan keuntungan kepada perusahaan dalam hal efisiensi biaya.

2.1.4 Lean Software Development

Lean Software Development adalah sebuah metode *agile* yang memiliki prinsip mengeliminasi hal-hal yang tidak diperlukan dan mengoptimasi waktu pengembangan serta sumber daya. Tidak seperti metode konvensional, *Lean Software Development* memberikan *guideline* untuk membuat produk dengan fitur seminimal mungkin yang biasa disebut *Minimum Viable Product* (MVP) yang bertujuan untuk menguji hipotesis pengembang terhadap user (MARTYAN, 2022).

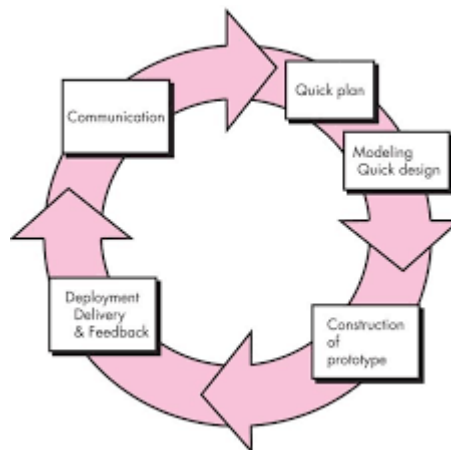
Lean memiliki prinsip penelitian dasar dalam pengembangannya yang terdiri dari tujuh bagian yaitu :

- 1) *Eliminate Waste*. Dalam penelitian ini hanya fitur-fitur inti yang akan digunakan agar tidak terjaidnya pemborosan dalam waktu. Fitur yang tidak memberikan tambahan nilai pada pengguna akan dihindari sejak tahap perancangan.
- 2) *Build Quality In*, kualitas sistem akan dijaga dimulai dari pengembangan hingga pengujian pada penelitian yang dilakukan. Agar penelitian dapat berjalan sesuai kebutuhan dan kesalahan dapat diminimalisir sejak awal.
- 3) *Create Knowledge*, dilakukan oleh tim pengembangan dengan mengumpulkan masukan dari pengguna.
- 4) *Defer Commitment* selama mungkin, untuk melakukan revisi berdasarkan semua masukan yang diberikan.
- 5) *Deliver Fast*, prinsip ini digunakan dalam pembuatan sistem sehingga sistem dapat dikirimkan dengan cepat kepada pelanggan. Agar pengguna dapat melihat sistem dan memberi masukan secepat mungkin.
- 6) *Respect People*, prinsip ini dilakukan untuk sesama anggota tim agar dapat membangun hubungan yang baik dan dapat menyelesaikan penelitian yang dilakukan.
- 7) *Optimize The Whole*, dilakukan ketika semua proses penelitian telah selesai dilakukan. Dan juga memastikan seluruh proses input dan output berjalan dengan lancar.

2.1.5 Prototyping

Model *prototyping* adalah cara cepat untuk mendapatkan informasi tentang kebutuhan pengguna tertentu yang berfokus pada menyajikan elemen perangkat lunak yang dapat dilihat oleh pengguna, untuk mengidentifikasi kebutuhan pengembangan perangkat lunak (Titania Pricillia & Zulfachmi, 2021). Metode yang digunakan pada penelitian ini adalah *prototype*. Metode *prototyping* akan menghasilkan prototipe sistem untuk membantu pengembang dan pengguna

berinteraksi selama proses pengembangan sistem informasi. Agar proses pembuatan prototipe berhasil, aturan harus ditetapkan pada tahap awal. Pengembang dan pengguna harus yakin bahwa prototipe dibangun untuk memenuhi kebutuhan awal. Sampai uji coba dilakukan secara bersamaan dengan proses pengembangan, prototipe akan dihilangkan atau ditambahkan sesuai dengan perencanaan dan evaluasi pengembang (Firnas Adani et al., 2022).



Gambar 2.2 Tahapan Model Prototype

Pada Gambar 2.2 (Kurniawan et al., 2023) pprototyping memiliki tahapan metode dalam pengembangannya yang terdiri dari lima tahap yaitu :

1. *Communication*

Pada tahap ini dilakukan wawancara untuk mengumpulkan kebutuhan yang akan identifikasi dan akan dimasukkan selama pembuatan sistem berlangsung. Dan juga di tahap ini akan menggabungkan dengan prinsip *Lean Software Development* yaitu prinsip *respect people & create knowledge*. Karena dalam wawancara membutuhkan komunikasi untuk memahami kebutuhan intinya serta kita dapat membangun pengetahuan baru agar pengembangan jelas sejak awal.

2. *Quick Plan*

Setelah mendapatkan kebutuhan dari pengguna, akan dilanjutkan untuk pembuatan rencana pembuatan sistem yang sesuai kebutuhan. Di tahap ini akan menggabungkan dengan prinsip *Lean Software Development* yaitu prinsip *eliminate waste & defer commitment*. Karena perencanaan dilakukan

dengan singkat yang hanya pembahasan fitur-fitur penting tanpa adanya pemborosan.

3. *Modeling Quick Design*

Pada tahap ini dilakukan perancangan model dengan rencana yang sudah di analisis berdasarkan kebutuhan yang dibutuhkan. Dan juga pada tahap ini akan menggabungkan dengan prinsip *Lean Software Development* yaitu prinsip *deliver fast & build quality in*. Karena pada tahap dalam pembuatan rancangan model dibutuhkan kualitas yang bagus.

4. *Construction of Prototype*

Pada tahap ini, dilakukan pembuatan rancangan aplikasi berdasarkan kebutuhan yang sudah dikumpulkan. Pembuatan ini ditujukan untuk menerima komentar dari pemilik sehingga kita dapat mengetahui rancangan dibuat sudah sesuai atau belum. Serta akan dilakukan penggabungan dengan prinsip *Lean Software Development* yaitu prinsip *build quality in & deliver fast*. Karena dalam pembuatan modelnya dibutuhkan kualitas yang sesuai agar dapat masuk ke tahap pengujian.

5. *Deployment Delivery & Feedback*

Pada tahap ini, dilakukan pengembangan prototipe yang sudah dibuat dan diimplementasikan ke dalam bahasa pemrograman untuk merancang sistem tersebut. Lalu akan dilakukan pengujian pada prototipe untuk menguji kelayakan dari sistem tersebut. Dan juga pada tahap ini akan menggabungkan prinsip *Lean Software Development* yaitu prinsip *create knowledge & optimize the whole*. Karena dalam pengujian dilakukan kita dapat menerima masukan baru dan mendapatkan pengetahuan yang berguna bagi pembuat.

2.1.6 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman skrip yang bersifat *open source* dan secara luas digunakan untuk pengembangan *web*. PHP memungkinkan pengembang menyisipkan kode langsung ke dalam HTML, sehingga dapat menghasilkan halaman web yang dinamis secara efisien. Kode PHP dijalankan di sisi *server* dan menghasilkan output dalam bentuk HTML yang

dikirim ke browser pengguna, tanpa memperlihatkan kode sumber aslinya. PHP memiliki keunggulan dalam kemudahan dalam penggunaan bagi pemula serta fleksibilitas tinggi bagi pengembang profesional. Selain digunakan untuk aplikasi *web*, PHP juga dapat dijalankan melalui *command line* dan memiliki kemampuan untuk menghasilkan berbagai jenis file seperti PDF, gambar, XML, dan JSON. PHP mendukung berbagai sistem operasi seperti Windows, Linux, dan macOS serta kompatibel dengan banyak *web server*. Salah satu kekuatan utama PHP adalah dukungannya terhadap berbagai jenis *database*, seperti MySQL, PostgreSQL, dan SQLite, yang membuatnya sangat dalam membangun aplikasi *web* berbasis data (N. Putri, 2023).

2.1.7 MySQL

MySQL merupakan sistem manajemen basis data relasional (RDBMS) bersifat *open-source* yang dikenal karena keandalan dan daya tahannya. Dalam konteks manajemen data secara modern, MySQL berperan penting dalam fondasi infrastruktur data yang membangun berbagai aplikasi dan sistem, dengan fokus pada efisiensi dan kinerja. Keunggulan utama MySQL tidak hanya terletak pada sifat terbukanya, tetapi juga dalam membuat desain *database*. Desain *database* MySQL yang baik mencakup aspek penting seperti pemilihan tipe data, normalisasi, pengurutan, dan pemilihan penyimpanan yang sesuai. Dengan pendekatan desain yang cukup baik, MySQL mampu menangani volume data besar serta beban kerja pengguna yang tinggi secara efisien (Soufitri, 2023).

2.1.8 Laravel

Laravel adalah sebuah *framework* PHP yang dirilis di bawah lisensi MIT dan dibangun dengan konsep MVC (*Model-View-Controller*). Laravel merupakan *framework* pengembangan website berbasis PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan. Selain itu, Laravel juga bertujuan untuk meningkatkan pengalaman dalam bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas, dan efisien (Malini et al., 2022).

2.1.9 Blackbox Testing

Metode *Blackbox Testing* adalah sebuah teknik pengujian fungsional yang

dirancang berdasarkan informasi dari spesifikasi sistem tanpa memerlukan pengetahuan tentang struktur internal sistem tersebut. Dalam konteks pengujian perangkat lunak, *Blackbox Testing* juga dikenal sebagai pengujian fungsional yang bertujuan memastikan bahwa fungsionalitas sistem berjalan sesuai dengan yang diharapkan tanpa harus mengetahui detail implementasi internal sistem (Zen et al., 2024).

Pengujian ini dilakukan dengan merancang *test case* berdasarkan spesifikasi sistem untuk memverifikasi apakah sistem berperilaku sesuai dengan yang diinginkan. Selain itu, pengujian *Blackbox* juga digunakan untuk memastikan bahwa sistem telah sesuai dengan spesifikasi dan rancangan yang telah ditetapkan. (Zen et al., 2024). Dalam konteks pengujian sistem informasi, pada tabel 2.1 terdapat beberapa contoh pertanyaan yang diajukan dalam metode blackbox testing meliputi:

Tabel 2.1 Contoh Pertanyaan BlackBox Testing

NO	Pertanyaan
1	Apakah sistem dapat menerima <i>username</i> dan <i>password</i> yang valid serta mengarahkan pengguna ke halaman <i>dashboard</i> sesuai perannya?
2	Apakah sistem akan menampilkan pesan kesalahan yang sesuai saat pengguna memasukkan <i>username</i> atau <i>password</i> yang salah?
3	Apakah pengajuan PO yang dilakukan oleh admin dengan data valid akan langsung muncul pada daftar PO admin dan pemilik?
4	Apakah sistem mampu menampilkan status terkini dari pengajuan yang dilihat oleh mandor?
5	Apakah progress proyek yang ditambahkan oleh mandor akan tampil di halaman list pengajuan?

2.1.10 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan pengujian yang dilakukan oleh end-user dimana user tersebut adalah staff/karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan atau fungsinya. Setelah dilakukan sistem *testing*, *acceptance testing* menyatakan bahwa sistem perangkat lunak memenuhi persyaratan (Supriatna, 2018).

Menurut (Supriatna, 2018), dalam penggunaan UAT memiliki beberapa manfaat yaitu:

- 1) Meningkatkan kepercayaan klien tentang potensi perangkat lunak untuk memenuhi persyaratan.
- 2) Melalui identifikasi cacat, pengujian memastikan bahwa perangkat lunak stabil dan dalam kondisi yang bisa diterapkan.
- 3) Kepuasan klien meningkat, karena mereka lebih yakin bahwa sistem sudah memenuhi persyaratan.
- 4) Mendapatkan sistem yang sesuai dengan spesifikasi fungsional sistem.

Dalam pelaksanaannya, pengujian UAT berfokus pada apakah sistem telah siap digunakan dari sudut pandang pengguna. Pada tabel 2.1 terdapat beberapa contoh pertanyaan yang diajukan dalam *User Acceptance Testing* antara lain:

Tabel 2.2 Contoh Pertanyaan *User Acceptance Testing* (UAT)

NO	Pertanyaan
1	Apakah pengguna dapat melakukan <i>login</i> dengan akun yang valid dan diarahkan ke halaman <i>dashboard</i> sesuai perannya?
2	Apakah sistem menolak <i>login</i> saat pengguna memasukkan <i>username</i> atau <i>password</i> yang salah, dan menampilkan pesan kesalahan yang sesuai?
3	Apakah admin dapat membuat pengajuan PO dan data tersebut muncul dalam daftar pengajuan milik admin maupun pemilik proyek?
4	Apakah pemilik proyek dapat memperbarui status PO dan perubahan tersebut tercermin kembali di halaman admin?
5	Apakah mandor dapat melihat status pengajuan barang serta menambahkan progress proyek secara mandiri?

2.1.11 Usability Testing

Usability Testing adalah metode pengujian yang menilai seberapa mudah dan efektif sebuah antarmuka sistem digunakan oleh pengguna. Menurut (Setiawan et al., 2022) indikator pada usability juga digunakan untuk mengukur seberapa puas pengguna dalam menggunakan teknologi, aplikasi, atau produk tersebut untuk mencapai goals atau tujuan, dalam hal ini ukuran keberhasilannya dilihat dari seberapa baik sebuah aplikasi atau teknologi dalam memberikan kualitas layanan kepada pengguna. Dalam usability testing terdapat 5 indikator yang bisa digunakan untuk mengetahui kualitas website yang digunakan oleh pengguna. Menurut (Setiawan et al., 2022) dalam usability testing terdapat kelima indikator sebagai berikut.

1. *Learnability*: indikator pada *usability* yang digunakan untuk mengetahui seberapa mudah pengguna dalam mempelajari *website* yang digunakan dalam memenuhi *task* yang ada.
2. *Memorability*: indikator yang digunakan untuk mengetahui sejauh apa kemudahan pengguna mengingat cara menggunakan *website* apabila setelah sekian lama tidak menggunakannya.
3. *Efficiency*: indikator yang digunakan untuk mengetahui seefisien apa para pengguna dalam melakukan beberapa task yang tersedia pada aplikasi.
4. *Errors*: indikator yang dipakai untuk mengetahui seberapa banyak kesalahan yang dilakukan oleh pengguna ketika menggunakan *website*, dan berkaitan dengan cara pengguna dalam memperbaiki kesalahan.
5. *Satisfaction*: indikator yang menjelaskan tingkat kepuasan pengguna dalam menggunakan *website*.

Pengujian dilakukan dengan 5 indikator dan terdapat 5 skala dari sangat tidak setuju hingga sangat setuju yang dapat dilihat di Tabel 2.3 .

Tabel 2.3 Skala Penilaian

Skala	Keterangan
1	Sangat tidak setuju
2	Tidak setuju
3	Netral
4	Setuju
5	Sangat Setuju

Skor per indikator :

$$\text{Skor Indikator} = \frac{\sum \text{jumlah pertanyaan}}{\text{nilai seluruh pertanyaan indikator}}$$

Skor usability keseluruhan :

$$\text{Skor Usability Total} = \frac{\sum \text{Semua Nilai}}{20}$$

Penilaian skor dapat dilihat dari tabel berikut ini.

Tabel 2.4 Implementasi Nilai

Rentang Skor	Interpretasi
4.21 – 5.00	Sangat Baik / Sangat Mudah digunakan

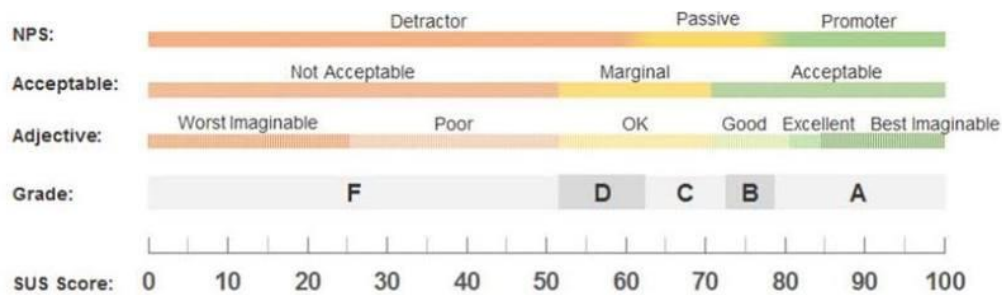
Rentang Skor	Interpretasi
3.41 – 4.20	Baik
2.61 – 3.40	Cukup
1.81 – 2.60	Kurang
1.00 – 1.80	Sangat Buruk

Pertanyaan dibawah ini merupakan contoh pertanyaan yang akan dilakukan setelah pengembangan sistem selesai dilakukan yang dapat dilihat di Tabel 2.5.

Tabel 2.5 Contoh Pertanyaan *Usability Testing*

NO	Pertanyaan	Skala				
		STS	TS	N	S	SS
1	Apakah tampilan antarmuka sistem informasi pengelolaan barang mudah dipahami?					
2	Apakah antarmuka sistem informasi pengelolaan barang nyaman digunakan dan tidak membosankan?					
3	Apakah Anda mudah mengingat kembali cara menggunakan menu setelah tidak memakai sistem beberapa waktu?					
4	Ketika mengoperasikan antarmuka sistem informasi pengelolaan barang, apakah terdapat <i>error</i> yang mengganggu kinerja sistem?					
5	Apakah sistem informasi pengelolaan barang memudahkan dalam mengelola pengajuan dan barang?					

Setelah seluruh jawaban responden dihitung dan diperoleh nilai *usability*, hasil pengujian kemudian diinterpretasikan berdasarkan kategori skor *System Usability Scale* (SUS). Interpretasi ini bertujuan untuk mempermudah peneliti dalam mengetahui tingkat penerimaan, kualitas dan penilaian subjektif terhadap sistem yang dikembangkan. Semakin tinggi skor SUS yang diperoleh, maka semakin baik tingkat *usability* sistem berdasarkan persepsi pengguna (Bangor et al., 2009).



Gambar 2.3 Ketentuan Penilaian Skor SUS

2.1.12 Unit Testing

Unit Testing merupakan metode pengujian perangkat lunak yang fokusnya pada bagian terkecil dari kode seperti fungsi, kelas atau modul. Untuk memastikan setiap unitnya dapat bekerja sesuai dengan logika yang sudah dirancang. Unit testing juga mendukung prinsip Build Quality In sehingga dapat mengurangi pemborosan yang muncul pada akhir pengembangan sistem. Berdasarkan Kulkarni (2020), *unit testing* merupakan bagian dari *automated testing* yang memiliki bagian dalam *Continuous Integration (CI)*. Setiap perubahan kode selalu diuji secara otomatis agar pengerjaan dapat berjalan lancar dan kesalahan dapat terdeteksi lebih awal, sehingga kualitas perangkat lunak tetap terjaga (Soares et al., 2022) .

Tabel 2.6 Contoh Pertanyaan *Unit Testing*

No	Kelas Uji	Fungsi yang diuji	Skenario Uji
1	<i>LoginController</i>	<code>authenticate()</code>	Input <i>username</i> dan <i>password</i> valid
2	<i>LoginController</i>	<code>authenticate()</code>	Input <i>username</i> dan <i>password</i> tidak valid
3	<i>POController</i>	<code>Store()</code>	Input data PO valid
4	<i>POController</i> <code>updateStatus()</code>	<code>updateStatus()</code>	Admin mengubah status pengajuan PO
5	<i>KonstruksiController</i>	<code>Store()</code>	Input data konstruksi valid

2.2 Penelitian Terdahulu

Penelitian ini dilakukan tidak terlepas dari hasil penelitian sebelumnya yang dilakukan sebagai data pembanding. Hasil penelitian yang dijadikan bahan pembanding tentunya memiliki keterkaitan dengan topik penelitian yang akan dilakukan, yaitu Rancang Bangun Sistem Informasi Pengelolaan dan Monitoring

Barang Lapangan pada Proyek Konstruksi dengan menggunakan metode *Lean Software Development* yang berfokus pada fitur-fitur penting.

Penelitian oleh Pohan et al. (2024), mengembangkan aplikasi berbasis web untuk keluhan konsumen terhadap sample hasil produksi benang. Metode yang digunakan adalah framework Laravel dengan metode *Lean Software Development* (LSD) yang dikombinasikan dengan pendekatan *Agile*. Aplikasi ini dibuat untuk meningkatkan efisiensi dalam penanganan keluhan di PT. Asia Pacific Fibers, Tbk., dengan menghilangkan proses-proses yang masih manual, lambat dan kurang terstruktur. Aplikasi ini dirancang melalui tahapan identifikasi kebutuhan, analisis, perancangan, implementasi, hingga pengujian dengan metode *Blackbox Testing* dan *User Acceptance Testing* (UAT). Berdasarkan hasil pengujian, aplikasi web ini diuji dengan *User Acceptance Testing* (UAT) mendapatkan penerimaan pengguna sebesar 85% dari skala 100, yang menunjukkan aplikasi ini layak digunakan.

Pada penelitian oleh Leipary et al. (2024), mengembangkan aplikasi sistem manajemen dosir bernama SMILE yang berfungsi untuk mengelola data kepegawaian di BPS Provinsi Maluku. Metode yang digunakan adalah *Lean Software Development* yang diikuti dengan bahasa pemrograman *PHP*, *HTML*, *CSS* dan pengujiannya menggunakan *Behavior Driven Development* (BDD). Hasil pengujian menunjukkan bahwa aplikasi SMILE berhasil diuji dengan melakukan *testing* yang berfokus pada *software behavior* tanpa ada *bug* atau *error*, yang berarti aplikasi ini telah memenuhi kriteria dari kualitas yang diinginkan dan memiliki kualitas yang baik.

Pada penelitian oleh Jayadi (2023), mengimplementasikan metode *Lean Software Development* dalam pengembangan aplikasi *client-server* untuk proses *import data Excel* ke *MySQL* secara dinamis. Metode yang digunakan adalah *Lean Software Development*. Dari hasil yang diberikan, aplikasi ini dapat mencapai rata-rata *import data* sebanyak 20000 baris dan 7 kolom hanya memerlukan 1,13 detik, yang berarti dalam penerapan *Lean Software Development* dapat meningkatkan efisiensi dan mengurangi waktu pengembangan.

Pada penelitian oleh A. S. Putri & Megawaty (2023), mengembangkan aplikasi *booking online* perjalanan wisata pada Sartana Tour and Travel berbasis *web* yang berfungsi untuk memudahkan pelanggan dalam melakukan pemesanan

paket wisata secara *online* dan membantu perusahaan dalam mengelola data *booking* pelanggan secara efektif. Metode yang digunakan adalah *Lean Software Development* (LSD) dan perancangan aplikasi menggunakan pemodelan *UML*. Hasil pengujian menunjukkan bahwa aplikasi ini berhasil diuji dengan melakukan uji pada setiap fitur dan evaluasi secara berulang, yang berarti aplikasi ini dapat digunakan secara langsung dan memberikan kemudahan dalam proses pemesanan. Perbandingan dengan penelitian terdahulu dapat dilihat di Tabel 2.7

Tabel 2.7 Perbandingan Penelitian Terdahulu

Nama	Judul	Platform	Metode	Hasil Studi Kasus
Sry Dhina Pohan dkk. (2024)	Implementasi <i>Lean Software Development</i> Pada Aplikasi Keluhan Sample Hasil Produksi Benang	<i>Web-based application</i>	<i>Lean Software Development & Agile Software Development</i>	Dapat mengurangi waktu proses penanganan keluhan dan memahami tingkat pemakaian oleh pengguna
Harfely Leipary dkk. (2024)	Rancang Bangun Aplikasi SMILE Menggunakan Metode <i>Lean Software Development</i>	<i>Web-based application</i>	<i>Lean Software Development</i>	Mengelola data kepegawaian di BPS Provinsi Maluku.
Puguh Jayadi (2023)	Mengimplementasikan metode <i>Lean Software Development</i> dalam pengembangan aplikasi client-server	<i>Web-based client-server application</i>	<i>Lean Software Development</i>	Meningkatkan efisiensi dan mengurangi waktu pengembangan dalam import data
Anaisti Sartana Putri dan Megawaty (2023)	Aplikasi Booking Online Perjalanan Wisata Pada Sartana Tour Dan Travel Berbasis Web Menggunakan Metode <i>Lean Development</i>	<i>Web-based application</i>	<i>Lean Development</i>	Memberikan kemudahan pemesanan paket wisata online pada pelanggan dan memudahkan perusahaan dalam mengelola data
Usulan Penelitian	Sistem Informasi Pengelolaan Dan Monitoring Barang Lapangan Pada Proyek Konstruksi Berbasis	<i>Web-based application</i>	<i>Lean Software Development</i>	dapat melaksanakan pengelolaan dan monitoring barang dengan

Nama	Judul	<i>Platform</i>	Metode	Hasil Studi Kasus
	Web Dengan Menggunakan Metode <i>Lean Software Development</i>			sistem serta dapat menghilangkan pemborosan barang.