

Bab II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori mencakup serangkaian konsep dan prinsip yang relevan dengan proyek akhir, berfungsi sebagai dasar ilmiah untuk mendukung analisis dan pemecahan masalah yang dihadapi. Landasan ini menyediakan kerangka acuan yang sistematis dalam merumuskan solusi, serta menjadi pedoman dalam menerapkan metodologi yang tepat untuk mencapai tujuan proyek akhir secara terstruktur dan terukur.

2.1.1 PT. Perkebunan Nusantara IV Regional III

PT. Perkebunan Nusantara IV (PTPN IV) yang setelah restrukturisasi dikenal dengan nama *PalmCo*, merupakan bagian dari *Subholding* PT. Perkebunan Nusantara III (Persero). Perusahaan ini berfokus pada komoditas utama kelapa sawit dan dibentuk melalui proses penggabungan antara PTPN V, VI, dan XIII ke dalam PTPN IV, yang menjadi entitas yang bertahan. PTPN IV yang selanjutnya disebut "Perusahaan", pada awalnya merupakan Badan Usaha Milik Negara yang didirikan berdasarkan Peraturan Pemerintahan Republik Indonesia (PP) No. 10 tahun 1999 tanggal 14 Februari 1996 tentang Penyetoran Modal Negara Republik Indonesia untuk pendirian perusahaan. Pada awalnya merupakan konsolidasi proyek-proyek pengembangan kebun eks PT. Perkebunan (PTP) II, PTP IV, dan PTP V di Provinsi Riau.

Anggaran Dasar Perusahaan telah mengalami beberapa kali perubahan. Perubahan pada tahun 2014 sejalan dengan terbitnya Peraturan Pemerintahan No. 72 tahun 2014 tentang Penambahan Penyertaan Modal Negara Republik Indonesia ke dalam Modal Saham Perusahaan Perseroan (Persero) PT. Perkebunan Nusantara III yang mengalihkan 90% saham PTPN V dari milik negara menjadi milik PTPN III. Perusahaan struktur saham ini merubah status Perusahaan dari BUMN menjadi Anak Perus dimulai pada era perusahaan Holding BUMN Perkebunan dengan PTPN III sebagai *Champion*. Untuk mengelola komoditi kelapa sawit, perusahaan memiliki 12 unit Pabrik Kelapa Sawit (PKS) dengan kapasitas olah terpasang sebesar 570 ton TBS per jam dengan hasil olahan berupa minyak sawit dan inti sawit. Kemudian untuk mengolah lanjut komoditi inti sawit, perusahaan memiliki 1 unit Pabrik *Palm Kernel Oil* dengan kapasitas terpasang sebesar 400 ton inti sawit/hari dengan hasil olahan berupa *Palm Kernel Oil* (PKO) dan *Palm Kernel Meal* (PKM).



Gambar 2. 1 Kantor PTPN IV Regional III

Pengelolaan areal tanaman saat ini memasuki peralihan dari siklus tanaman pertama (Gen-1) menuju siklus tanaman kedua (Gen-2). Siklus pertama dimulai pada era tahun 1980-an melalui proyek-proyek pengembangan kebun eks PT. Perkebunan (PTP) II, IV dan V di Provinsi Riau. Peralihan dari Gen-1 menuju Gen-2 telah dimulai sejak tahun 2003 yang ditandai dengan *replanting* areal-areal tanaman usia tua renta yang sudah menurun nilai ekonomis produksinya. Fase peralihan Gen-1 ke Gen-2 ini diperkirakan tuntas pada tahun 2017. Pada saat itulah, seluruh tanaman Gen-2 yang diharapkan lebih produktif dibandingkan Gen-1, sebagai buah inovasi berlanjut di bidang budidaya tanaman.

2.1.2 Sistem Informasi

Sistem Informasi merupakan kombinasi dari teknologi informasi dan aktivitas manusia yang menggunakan teknologi itu untuk mendukung operasi dan manajemen organisasi (*Chambers, 1964*). Pada dasarnya, sistem informasi mengintegrasikan berbagai komponen seperti perangkat keras, perangkat lunak, infrastruktur, data, dan pengguna untuk mengumpulkan, memproses, menyimpan, serta mendistribusikan informasi yang berguna dalam pengambilan keputusan dan pengelolaan organisasi (*Chambers, 1964*).

Sistem informasi berperan penting dalam mengintegrasikan berbagai komponen teknologi informasi seperti perangkat keras, perangkat lunak, infrastruktur, dan database dengan aktivitas manusia sebagai penggunaanya (*Hussein et al., 2023*). Integrasi ini memungkinkan organisasi untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi secara efektif dan efisien (*Malongo et al., 2019*). Informasi yang diolah oleh sistem informasi sangat berguna dalam mendukung proses pengambilan keputusan yang tepat serta membantu pengelolaan organisasi

secara keseluruhan (Sari & Priantinah, 2019).

2.1.3 Data dan Informasi

Data merupakan kumpulan fakta mentah yang belum melalui proses pengolahan sehingga belum memiliki makna tertentu. Data dapat berupa angka, teks, simbol, maupun hasil pencatatan yang diperoleh dari proses observasi atau pengukuran. Menurut Turban et al., data merupakan bahan dasar yang nantinya akan diolah menjadi informasi yang lebih bermakna. Contoh data antara lain nama, tanggal, jumlah, maupun catatan transaksi tertentu.

Sementara itu, informasi merupakan hasil pengolahan data yang telah memiliki makna, nilai, dan manfaat bagi pengguna, khususnya dalam mendukung proses pengambilan keputusan. Davis dan Olson menyatakan bahwa informasi adalah data yang telah diproses sehingga menjadi lebih berguna dan dapat memberikan pemahaman bagi penerimanya. Informasi yang baik umumnya memiliki beberapa karakteristik, seperti relevan, akurat, tepat waktu, lengkap, dan mudah dipahami.

Dalam sistem informasi, data menjadi komponen utama yang akan diproses menggunakan teknologi maupun algoritma tertentu untuk menghasilkan informasi yang bermanfaat. Sebagai contoh, data mengenai pinjaman dan tanggal jatuh tempo pembayaran dapat diolah menjadi informasi berupa notifikasi pengingat pembayaran kepada mitra. Hal tersebut menunjukkan bahwa pengelolaan data yang baik dan terstruktur sangat penting agar informasi yang dihasilkan dapat mendukung aktivitas operasional serta membantu proses pengambilan keputusan dalam organisasi. Oleh karena itu, data dan informasi memiliki hubungan yang saling berkaitan, dimana data berperan sebagai dasar dalam menghasilkan informasi yang bermakna.

2.1.4 Framework Laravel

Laravel adalah salah satu *framework PHP* yang dirancang untuk memudahkan pengembangan aplikasi *web* dengan menyediakan fitur-fitur canggih dan *sintaks* yang elegan. *Framework* ini menggunakan arsitektur *MVC (ModelView-Controller)*, yang memisahkan logika aplikasi antarmuka pengguna, dan proses bisnis. Pendekatan ini membuat kode lebih terstruktur dan mudah dipelihara. Salah satu fitur andalan *Laravel* adalah *Blade Templating Engine*, yang memungkinkan pembuatan antarmuka pengguna dinamis dengan *sintaks* yang sederhana dan efisien. Selain itu, *Laravel* dilengkapi dengan *Eloquent ORM (Object Relational Mapping)*, yang mempermudah pengelolaan dan interaksi dengan database melalui model berbasis objek.

Laravel juga menawarkan sistem *routing* yang fleksibel untuk menangani berbagai permintaan *HTTP*, seperti *GET*, *POST*, dan lainnya. Fitur Artisan *CLI* (*Common Line Interface*) memudahkan pengembang dalam membuat dan mengelola komponen aplikasi, seperti model, controller, dan migrasi database dan *seeding* untuk mempermudah manajemen skema dan memasukkan data awal. *Middleware* di *Laravel* memungkinkan pengelolaan permintaan *HTTP* sebelum mencapai aplikasi, seperti autentikasi pengguna atau manipulasi data.

Beberapa fitur yang dimiliki oleh *Framework Laravel* menurut Aminudin (2015):

1. *Bundles, bundles* merupakan fitur pengemasan modular dengan berbagai *bundle* yang telah tersedia untuk digunakan dalam aplikasi.
2. *Eloquent ORM*, implementasi *PHP* lanjutan dari pola "*active record*" yang menyediakan metode internal untuk mengelola hubungan antara objek dan database.
3. *Application Logic*, fitur ini merupakan bagian dari aplikasi yang sedang dikembangkan, dan dapat diimplementasikan menggunakan *Controllers* atau sebagai bagian dari deklarasi *Route*.
4. *Reverse Routing*, fitur yang memungkinkan pengguna untuk mendefinisikan hubungan antara *Link* dan *Route*. Dengan demikian, jika terjadi perubahan pada *route*, secara otomatis *link-link* yang terhubung akan tetap relevan. Ketika pembuatan *Link* menggunakan nama-nama dari *Route* yang ada, *Laravel* secara otomatis akan menghasilkan *URI* yang sesuai tanpa perlu dilakukan secara manual.
5. *Restful Controllers*, memberikan opsi bagi pengembang untuk memisahkan logika dalam menangani permintaan *HTTP GET* dan *POST* sehingga meningkatkan kejelasan kode aplikasi.
6. *Class Auto Loading*, fitur yang memberikan kemampuan otomatis untuk memuat (*load*) *class-class PHP* tanpa perlu melakukan pemeriksaan manual terhadap jalur masuknya. Dengan adanya fitur ini, *class-class* yang dibutuhkan akan dimuat secara otomatis ketika dibutuhkan, sehingga mencegah *loading* yang tidak perlu.
7. *View Composers*, kode yang dapat dieksekusi ketika sebuah *View* dimuat.
8. *IoC Container*, memungkinkan pembuatan objek baru dengan menerapkan prinsip kontrol pembalikan (*Inversion of Control*) dengan pilihan contoh dan referensi objek baru sebagai *Singleton*.
9. *Migrations*, merupakan mekanisme yang menyediakan kontrol versi untuk skema *database*, sehingga pengguna dapat menyelaraskan perubahan basis kode aplikasi dengan kebutuhan perubahan dalam tata letak database.

10. *Unit Testing*, fitur ini memiliki banyak tes yang berguna untuk mendeteksi dan mencegah regresi (kemunduran dalam kualitas perangkat lunak). Unit testing dalam *Laravel* dapat dijalankan melalui perintah baris perintah "*artisan command-line*".

11. *Automatic Pagination* merupakan fitur yang memudahkan penerapan halaman. Fitur ini menggantikan metode manual dengan cara otomatis yang sudah terintegrasi secara langsung ke dalam *Laravel*. Pengguna tidak perlu mengatur halaman secara manual, karena *Laravel* akan melakukan proses paginasi secara otomatis.

12. Keunggulan lain dari *Laravel* adalah fitur autentifikasi dan otorisasi bawaan yang kuat, penjadwalan tugas otomatis dengan *Task Scheduler*, serta dukungan penuh untuk pengembangan *RESTful API*. *Laravel* juga mempermudah integrasi dengan layanan pihak ketiga, seperti layanan pembayaran, email, dan notifikasi. Dengan dokumentasi yang lengkap dan komunitas pengembang yang besar, *Laravel* menjadi unggul bagi pengembang untuk membangun aplikasi *web modern*.

2.1.5 Laragon

Laragon adalah sebuah *universal development environment* yang dirancang khusus untuk PHP, *Python*, dan terisolasi serta mudah dipakai. *Laragon* memiliki kelebihan diantaranya yaitu bisa diakses menggunakan *localhost*, bisa dipindahkan tanpa merusak sistem, tidak mempengaruhi computer lokal, dan cocok digunakan untuk pengembangan *web*. Keunggulan utama *Laragon* terletak pada kecepatannya, kemudahan instalasi, serta kemampuannya untuk berjalan secara mandiri (*self-contained*) tanpa memengaruhi sistem utama computer pengguna.

Salah satu fitur unggulan *laragon* adalah kemampuannya dalam membuat *virtual host* secara otomatis hanya dengan satu klik, yang sangat berguna dalam pengembangan proyek *laravel* atau *framework PHP* lainnya. *Laragon* juga menyediakan terminal emulator bawaan, serta integrasi dengan *package manager* seperti *Composer* dan *npm* yang sangat dibutuhkan dalam proyek modern berbasis *web*. *Laragon* juga mendukung berbagai versi *PHP*, sehingga *developer* dapat dengan mudah berpindah antar versi sesuai kebutuhan proyek. Selain itu, kemampuannya untuk berjalan secara *portable* menjadikannya pilihan ideal untuk *developer* yang bekerja lintas perangkat atau ingin menghindari instalasi lokal yang rumit.

Dalam perkembangan sistem informasi seperti Notiloan menggunakan *laravel*, *laragon* menjadi alat bantu yang sangat efektif karena mampu menyediakan lingkungan pengembangan

yang stabil, ringan, dan fleksibel. Hal ini mempercepat proses pembuatan dan pengujian sistem tanpa harus melakukan konfigurasi manual yang memakan waktu.

2.1.6 WhatsApp Gateway

WhatsApp merupakan aplikasi untuk saling berkirim pesan secara instan, dan untuk saling bertukar gambar, video, foto, pesan suara, dan dapat digunakan untuk berbagi informasi dan diskusi (Puspaningrum, Rohmawati, 2020). *API WhatsApp* memungkinkan pengguna untuk mengirim dan menerima pesan *WhatsApp* dalam program mereka sendiri menggambarkan socket *Web* atau *HTTP*. Hal ini dilakukan dengan menggunakan perpustakaan *WhatsApp-API* berbasis *PHP* untuk mengakses *WhatsApp*. *API WhatsApp* untuk berinteraksi dengan *system* yang dibangun walaupun proyek mereka tidak ditulis dalam bahasa *PHP* ataupun menggunakan *PHP* (Deus, Frdialto Mario, 2017).

Whatsapp Gateway merupakan sebuah sistem aplikasi yang digunakan untuk mengirimkan pesan melalui pihak ketiga ke perangkat pengguna otomatis melalui *platform WhatsApp*. Integrasi dengan *WhatsApp Gateway* memungkinkan penyampaian informasi kepada pengguna secara otomatis, seperti pengingat jatuh tempo pembayaran. Hal ini sangat efektif untuk mengurangi keterlambatan pembayaran pinjaman.

2.1.7 Task Scheduler

Task scheduler adalah komponen dari *Microsoft Windows* yang menyediakan kemampuan untuk menjadwalkan program atau *script* pada waktu yang telah ditentukan atau setelah waktu interval tertentu. *Scheduler task* berjalan setiap kali menjalankan *windows* dan menjalankan latar belakang, dan menjalankan setiap tugas yang telah dijadwalkan pada waktu menentukannya saat membuat tugas (Simamora, Yahya, and Akbar 2018).

Dengan aplikasi *Task Scheduler* ini, kamu bisa membuat jadwal aktifitasmu sehari-hari dan kamu juga mengaktifkan alarm agar kamu ingat dengan jadwalmu. Selain bisa menjadwalkan program yang bisa menjadwalkan program yang bisa kita jalankan setiap hari, program ini juga bisa mengirim *e-mail* secara otomatis. Program ini juga bisa mematikan komputer pada waktu yang telah ditentukan dan juga membuat pesan pengingat sehingga sekarang kamu tidak lagi akan melewatkan tugas yang sangat penting (Stallings 2009).

2.1.8 Agile

Metode *Agile software development* atau biasa disebut dengan *agile* adalah proses *iterative* dalam pembuatan sebuah perangkat lunak. *Agile* merupakan metode pengembangan yang cepat

karena proses utama dari metode pengembangan *agile* sendiri berfokus pada *design-codetest once day* (Lutfiani, dkk. 2020). Dalam pengembangan *Agile*, terdapat 7 macam model diantaranya *SCRUM*, *extreme programming (XP)*, *Kanban*, *Cystal Methodology*, *Dynamic Systems Development*, dan *Lean Software Development*.

Metode *agile* menjadi populer saat ini karena memberikan fleksibilitas kepada pengembangan untuk kembali ke tahap awal jika terjadi perubahan yang diperlukan. Metode ini adalah salah satu pendekatan dalam pengembangan perangkat lunak yang efektif dan adaptif. Berbeda dengan metode tradisional, metode *agile* tidak mengatur prosedur secara rinci tentang cara membuat model yang telah ditentukan (Larasati dkk. 2021).

Prinsip-prinsip yang diterapkan dalam pengembangan metodologi *agile* menurut Raharjana (2017) adalah:

1. *'First just right' process*, dengan hanya dalam pengembangan yang penting berdasarkan *best practice*.
2. *Continous testing and validation*, dengan hanya melakukan pengujian tidak hanya di akhir proses tetapi dari awal hingga akhir, serta dilakukan secara berkelanjutan.
3. *Consistent team collaboration*, kolaborasi antar anggota tim sangat penting dalam pengembangan menggunakan pendekatan *agile*. Tim melakukan pertemuan singkat secara berkala untuk membahas kemajuan masing - masing anggota dan hasil kerja tiap anggota dapat segera diketahui anggota lainnya.
4. *Rapid response to change*, dalam pendekatan *agile*, perubahan dianggap bagian dari proses pengembangan perangkat lunak, dengan kata lain perubahan akan ditangani bukan dihindari.
5. *Ongoing customer involvement*, dengan melibatkan pelanggan dalam proses pembuatan, tingkat penerimaan pelanggan akan lebih tinggi daripada jika hanya dilibatkan pada tahap *User acceptance test*.
6. *Frequent delivery of working software*, pembangunan perangkat lunak menggunakan pendekatan *agile* dibagi menjadi iterasi - iterasi dalam prosesnya. Tiap iterasi menghasilkan kode yang berfungsi atau modul perangkat lunak yang telah selesai dibuat dan diuji.

2.1.9 Kanban

Kanban merupakan pendekatan untuk memvisualisasikan alur kerja (*workflow*) dan pekerjaan yang harus dilakukan pada tim pengembang, dengan menggunakan teori antrian untuk mengontrol serta meningkatkan aliran nilai dengan mengarahkan perhatian pada siklus produksi.

Keuntungan menggunakan kanban yaitu proses komunikasi tim menjadi lebih interaktif, dimana dalam setiap pekerjaan atau proyek melibatkan pihak klien. Penggunaan kanban dalam penerapan sistem kerja dapat memaksimalkan bagi proses bisnis, seperti meningkatkan sistem alur kerja, mempersingkat waktu yang dibutuhkan dalam pekerjaan, meningkatkan nilai bagi klien (dos Santos et al., 2019).

Menurut Anderson dalam (dos Santos et al., 2019) kanban memiliki lima prinsip:

- 1) Visualisasikan alur kerja, papan adalah alat utama yang digunakan untuk memvisualisasikan dan kordinasi sesama kerja sama tim, dimana setiap kolom menunjukkan urutan aktivitas dan kartu merupakan fitur yang sedang dikerjakan.
- 2) Batasan pekerjaan yang sedang berjalan, ini merupakan cara untuk mengelola dan membatasi jumlah pekerjaan yang sedang berjalan serta pemberian tugas yang baru.
- 3) Pengukuran dan pengelolaan aliran kerja, statistik dan diagram yang berbeda dapat digunakan untuk memantau proses Kanban seperti siklus/waktu tunggu, ukuran antrian, dan diagram aliran kumulatif
- 4) Buat kebijakan proses secara eksplisit, kebijakan merupakan bagian penting untuk memastikan alur tercapai. Mereka menetapkan kondisi untuk membuat sistem terus bekerja. Misal cara menetapkan tugas dan aktivitas ke pengembang dan kapan pekerjaan dapat ditarik dari satu status ke status lainnya.
- 5) Gunakan model untuk mengenali peluang peningkatan: tiga model disarankan (i) Teori kendala, (ii) Bagian kumpulan ide dari *Lean Thinking* yang mengidentifikasi pemborosan biaya ekonomi, dan (iii) beberapa varian yang berfokus pada pemahaman dan pengurangan variabilitas.

7. Berikut tahapan penelitian menggunakan metode *Agile* dengan kerangka kerja *Kanban*:

1) *To Do*

8. Kolom ini merupakan daftar pekerjaan berbentuk *card task* yang harus dilakukan untuk mencapai tujuan proyek. Di dalam kategori ini terdapat beberapa poin diantaranya:

a) *Product Blocklog*

9. *Product Blocklog* merupakan rincian daftar tugas secara umum apa saja yang harus dikerjakan berdasarkan keahlian dari tim pengembang, seperti *card task* pembuatan *home page* yang terdapat pada *website*.

b) *Requirements*

10. *Requirements* merupakan kebutuhan detail suatu produk yang di dapat dari pengumpulan data. Dimana data didapatkan dari *user story* yang dimaksud ialah cerita detail dari pengguna perihal berjalannya proses sistem yang dibutuhkan, selanjutnya ada *feature* yang merupakan hasil rumusan cerita pengguna berupa rincian fitur – fitur untuk menunjang pengembangan aplikasi dan ada *Use Case* dimana rancangan proses berjalannya sistem digambarkan dalam bentuk *chart* pada *Unified Modelling Language* yang akan memudahkan tim pengembang dalam mengetahui alur dari berjalannya proses sistem yang akan dibangun.

c) *Design*

11. Tahapan ini merupakan perancangan dalam membangun sebuah sistem, dimana didalam tim pengembang terdapat divisi sesuai keahliannya seperti *ui/ux* yang akan membuat usul desain *interface* secara kreatif dan imajinatif, ada *front-end* yang akan mengubah hasil desain ke dalam tampilan *browser* serta merancang proses berjalannya sistem dalam bentuk *Unified Modelling Language*, selanjutnya *back-end* yang bertanggung jawab atas basis data sebuah sistem serta dirancang dalam bentuk *Entity Relationship Diagram*.

d) *Implementatior*

12. Tahapan ini merupakan kegiatan integrasi atau menghubungkan fitur – fitur yang sudah dibuat dengan basis data sebagai penyimpanan otomatis.

e) *Test*

13. Tahapan terakhir ini dilakukan pengujian dalam mencari kesalahan pada sistem. Proses ini akan mengevaluasi sistem baik secara manual maupun otomatis.

2) *In Progress*

14. Kolom ini merupakan sedang dikerjakannya tugas - tugas oleh tim pengembang. Dimana *card task* dari kolom *to do* pada papan *board* dipindah ke kolom *in progress* atau biasa disebut *WIP (work in progress)*.

3) *Done*

15. Kolom ini merupakan sudah diselesaikannya pekerjaan oleh tim pengembang, tugas - tugas sebelumnya yang berada pada kolom *work in progress* diimplementasikan lalu dilakukan *testing* jika berhasil maka dipindahkan *card task* ke kolom *done*. Jika ada pekerjaan yang tidak

bisa diselesaikan atau gagal saat dilakukan *testing*, maka *card task* akan di pindahkan ke kolom *to do* untuk didiskusikan dan mencari solusi permasalahan.

16. Sistem *kanban* diimplementasikan sebagai papan di dinding dengan kolom yang mewakili tahapan proses pengembangan berurutan. Kartu digunakan untuk mendeskripsikan pekerjaan atau tugas, yang dipindahkan melalui kolom pada papan. Penerapan *kanban* dalam pengembangan perangkat lunak memiliki tahapan kolom pengerjaan seperti tahapan spesifikasi atau *backlog* (spesifikasi), *development* (pengembangan), *testing* (pengujian), dan *deploy* (penerapan) (dos Santos et al., 2019).

2.1.10 Black Box Testing

Black Box Testing merupakan jenis pengujian yang berfokus pada spesifikasi fungsional perangkat lunak. Dalam metode ini, para pengujian dapat mendefinisikan berbagai kondisi input dan menguji program berdasarkan spesifikasi fungsionalnya (Mustaqbal et al., 2015).

Metode pengujian *Black Box Testing* merupakan salah satu pendekatan yang mudah digunakan karena hanya membutuhkan batas bawah dan batas atas dari data yang diharapkan. Jumlah estimasi data uji dapat dihitung berdasarkan jumlah *field entri* data yang akan diuji dan aturan *entri* yang berlaku. Dengan menggunakan metode ini, dapat diketahui apakah fungsionalitas masih dapat menerima masukan data yang tidak diharapkan yang menyebabkan data yang disimpan menjadi tidak valid (Cholifah et al., 2018).

2.1.11 Integration Testing

Integration Testing merupakan salah satu pengujian perangkat lunak dimana modul - modul yang sebelumnya berdiri sendiri digabungkan dan diuji sebagai sebuah kesatuan. Tujuan dari pengujian ini ialah untuk menggabungkan modul - modul unit sistem yang telah diuji dan membentuk sebuah struktur program sesuai dengan perancangan awal. Pengujian integrasi mencakup berbagai area yang menjadi perhatian saat mengintegrasikan modul-modul yang berbeda, antara lain:

1. Meninggalkan komponen perangkat lunak selama saling berinteraksi.
2. Berbagi data dan informasi antar modul dengan cara yang tepat.
3. *Compatibility*, untuk memastikan salah satu modul tidak akan mempengaruhi modul lain dari segi performansi dan fungsionalitas.
4. Isu persyaratan non-fungsional.

5. Pengujian Integrasi menguji atribut khusus dari perangkat lunak, yakni bagaimana komponen - komponen sistem berinteraksi sesuai dengan rencana sistem yang telah dibuat. Untuk itu, data uji harus difokuskan pada interaksi antar komponen serta menilai apakah integrasi yang telah dilakukan sesuai dengan tujuan perancangan sistem.

2.1.12 User Acceptance Testing

UAT merupakan salah satu metodologi paling inovatif untuk menghindari kegagalan pada proyek IT. Karenanya *User Acceptance Testing* dilakukan untuk mengumpulkan masukan dari calon pengguna dari aplikasi dan seseorang yang memiliki pengalaman dalam proses bisnis ketika diminta untuk menyelesaikan tugas terkait aplikasi. Pengujian *UAT* merupakan tahap akhir dari kegiatan pengujian untuk mengevaluasi kesiapan sistem sehingga akan melibatkan interaksi langsung antara pengguna dan sistem. Tujuan utamanya adalah untuk memverifikasi bahwa fitur - fitur berjalan sesuai dengan kebutuhan pengguna. Pengujian penerimaan pengguna harus mengizinkan pengguna untuk menentukan apakah aplikasi yang dibangun dapat diterima atau tidak. Berdasarkan penentuan itu dapat membuat pengguna merasa percaya diri bahawa aplikasi sudah sesuai dengan spesifikasi dan berperilaku sesuai dengan yang diharapkan. Dalam pengembangan perangkat lunak, terdapat tiga hal yang dapat terselesaikan dengan melakukan *UAT*, yaitu:

- 1) Membongkar logika bisnis atau fungsionalitas yang tidak ditemukan
- 2) Mengukur bagaimana sistem sesuai dengan kebutuhan pengguna.
- 3) Membatasi bagaimana sistem sudah diselesaikan.

2.2 Penelitian Terdahulu

Penelitian mengenai perancangan dan pengembangan sistem ini telah dilakukan oleh peneliti terdahulu. Oleh karena itu, diperlukan studi literatur untuk menghindari adanya kesamaan dengan penelitian sebelumnya. Beberapa penelitian dari lima tahun terakhir dijadikan sebagai referensi dalam penyusunan proyek akhir ini, antara lain sebagai berikut.

Penelitian pertama oleh Prasetio & Silalahi (2023). Penelitian ini bertujuan untuk merancang sistem e-administrasi berbasis *framework codeigniter* dengan Notifikasi *Whatsapp* untuk diaplikasikan pada Perumahan Griya Pratama. Hasil akhir yang akan dicapai adalah Perancangan Sistem E-Administrasi Berbasis *Framework Codeigniter* dengan Notifikasi *WhatsApp* yang dirancang untuk memudahkan proses administrasi khususnya dalam proses pendataan warga. RT terkait tidak perlu lagi mendatangi rumah warga satu persatu dan juga dalam pengadaan agenda

kegiatan salah satunya kegiatan gotong royong atau agenda kegiatan lainnya hanya perlu melakukan notifikasi pada fitur yang telah disediakan dalam sistem pada *website*.

Penelitian selanjutnya oleh Rampi Yusuf dkk. (2025). Penelitian terdahulu ini membuktikan bahwa penerapan sistem berbasis notifikasi otomatis mampu meningkatkan efisiensi dalam pengelolaan pembayaran serta meminimalkan keterlambatan. Studi di Pegadaian Gorontalo Sentral dan sektor lain menunjukkan bahwa penggunaan *WhatsApp API* sebagai pengingat pembayaran memberikan dampak positif dalam mengurangi risiko keterlambatan dibandingkan metode manual. Sistem otomatis ini memungkinkan pemberitahuan jatuh tempo dikirim secara *real-time*, sehingga pihak administrasi tidak perlu lagi melakukan pengecekan dan pengingat secara manual kepada pengguna. Sistem ini tidak hanya dapat mengotomatiskan pengelolaan data pinjaman, tetapi juga mengirimkan pengingat jatuh tempo secara otomatis melalui *WhatsApp*, sehingga proses administrasi menjadi lebih efisien dan risiko keterlambatan pembayaran dapat diminimalkan.

Penelitian selanjutnya oleh Dedi Satria (2023). Penelitian ini bertujuan untuk meningkatkan efisiensi dan efektivitas komunikasi dalam pelayanan akademik. Dalam perancangan sistem ini, *WhatsApp* digunakan sebagai media utama untuk menyampaikan notifikasi kepada mahasiswa dan staff administrasi terkait permohonan pelayanan dokumen akademik. Sistem informasi ini dibangun menggunakan pemrograman *PHP*, *Database*, *MySQL*, dan *WhatsApp API*. Penelitian ini menghasilkan beberapa data yaitu data mahasiswa, jenis pelayanan dan pelayanan. Sedangkan pada sisi riwayat pelayanan akademik per mahasiswa dan notifikasi kepada mahasiswa terkait status selesainya pelayanan dokumen akademik bagi mahasiswa. Berbasis Media Sosial *WhatsApp*, diharapkan sistem notifikasi ini saat memberikan kontribusi dalam meningkatkan kualitas pelayanan akademik dan memenuhi kebutuhan penggunanya.

Penelitian selanjutnya oleh Khoeriyah (2021). Penelitian ini bertujuan untuk mengetahui pemanfaatan layanan *whatsapp gateway* sebagai sistem notifikasi pinjaman (SINOPI). Metode yang digunakan dalam penelitian ini adalah metode kualitatif dengan analisis deskriptif. Teknik pengumpulan data yang dilakukan dalam penelitian ini adalah dengan wawancara. Penelitian ini menghasilkan temuan bahwa dengan adanya layanan ini setidaknya dapat meminimalisir keterlambatan pengembalian buku sebanyak 20 eksemplar per harinya. Dengan sistem ini, pemustaka dapat memperpanjang masa pinjaman secara otomatis tanpa harus datang ke perpustakaan. Namun, beberapa kendala masih ditemukan, seperti notifikasi yang tidak

tersampaikan akibat pemadaman listrik, nomor pemustaka yang tidak aktif, atau kuota internet yang habis. Secara keseluruhan, sistem ini meningkatkan efisiensi layanan perpustakaan dan berpotensi untuk diterapkan di berbagai sektor yang memerlukan sistem pengingat otomatis.

Penelitian selanjutnya oleh Makarim & Fadillah (2024). Penelitian ini mengembangkan sistem pelacakan dan pengingat otomatis peminjaman alat farmasi dengan memanfaatkan *Whatsapp gateway*. Sebelumnya, proses peminjaman yang dilakukan secara manual sering mengakibatkan keterlambatan pengembalian, sehingga mengganggu ketersediaan alat bagi pengguna lainnya. Untuk mengatasi permasalahan ini, sistem yang dirancang memberikan notifikasi otomatis kepada peminjam saat jatuh tempo pengembalian alat, sehingga meningkatkan kepatuhan terhadap aturan peminjaman. Pengembangan sistem dilakukan dengan metode *Waterfall*, melalui tahapan analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan. Pengujian menunjukkan bahwa penggunaan sistem ini mampu meningkatkan efisiensi waktu mencapai 67%. Penerapan *Whatsapp gateway* dalam sistem ini berhasil mempermudah administrasi peminjaman dan memastikan pengembalian alat tepat waktu.

Tabel 2 1 Penelitian Terdahulu

Penulis	Jenis Perbandingan		
	Metode	Fitur	Hasil
Prasetio & Silalahi (2023)	<i>Extreme Programing</i>	WhatsApp	Perancangan Sistem EAdministrasi Berbasis <i>Framework codeigniter</i> dengan Notifikasi WhatsApp yang dirancang untuk memudahkan proses adminisrasi khususnya dalam proses pendataan warga, RT terkait tidak perlu lagi mendatangi rumah warga satu persatu.

Penulis	Jenis Perbandingan		
	Metode	Fitur	Hasil
Rampi Yusuf dkk (2025)	Identifikasi Kebutuhan	WhatsApp	Dengan adanya sistem ini dapat membantu pengingat pembayaran nasabah dengan lebih akurat dan tepat waktu, mengotomatiskan proses pengiriman notifikasi pengingat melalui WhatsApp dan meningkatkan efisiensi pengelolaan pembayaran yang sebelumnya dilakukan secara manual.
Dedi Satria (2023)	<i>Software Developer Life Cycle (SDLC)</i>	WhatsApp	Implementasi form dan laporan yaitu form mahasiswa, form jenis pelayanan, form pelayanan.
Khoeriyah (2021)	Kualitatif dengan Analisis Deskriptif	WhatsApp	<i>SINOPI Whatsapp Gateway</i> meminimalisir keterlambatan pengembalian buku kurang lebih 20 eksemplar buku perharinya.
Makarim & Fadillah (2024)	<i>Waterfall</i>	WhatsApp	Dengan adanya sistem ini, memberikan pemberitahuan jika masa peminjaman alat laboratorium sudah berakhir atau jatuh tempo tanggal akhir peminjaman.

Berdasarkan beberapa penelitian terdahulu, dapat disimpulkan bahwa penggunaan notifikasi berbasis *WhatsApp* sudah banyak di terapkan dalam berbagai sistem untuk meningkatkan efisiensi informasi dan pengingat otomatis. Sama halnya dengan Sistem Notiloan menggunakan *Laravel Puppeteer Library*, seluruh penelitian tersebut menekankan pentingnya sistem pemberitahuan otomatis melalui *WhatsApp* untuk meningkatkan kecepatan waktu, efisiensi kerja, dan mengurangi keterlambatan dalam proses layanan yang pada akhirnya memperkuat relevansi dan efektifitas Sistem Notiloan dalam pengelolaan pinjaman di PTPN IV Regional III.