

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori memuat kajian pustaka yang menjadi dasar ilmiah dalam pengembangan sistem pada penelitian ini. Teori-teori yang diuraikan mencakup konsep teknologi, metodologi, dan tools yang digunakan secara langsung dalam proses perancangan dan pembangunan aplikasi mobile Bank Jatah Indonesia.

2.1.1 Bank Jatah Indonesia

Bank Jatah Indonesia merupakan usaha yang bergerak dalam pengelolaan minyak jelantah dari masyarakat. Saat ini, proses bisnis di Bank Jatah Indonesia masih berjalan secara manual, melibatkan tiga peran utama, yaitu Admin (Unit Utama), Unit Bisnis (mitra di beberapa kelurahan), dan Nasabah (masyarakat penyeter minyak jelantah). Admin berperan sebagai pengelola utama yang mengatur distribusi dana dan validasi setoran. Unit Bisnis berjumlah 57 yang tersebar di Provinsi Riau, dengan tugas menerima minyak jelantah dari nasabah yang berjumlah 620 Nasabah, menyeterkannya ke unit utama, serta mengambil uang dari unit utama untuk kemudian diserahkan kepada nasabah. Sementara itu, nasabah merupakan masyarakat umum yang dapat menyeterkan minyak jelantah baik langsung ke unit utama maupun melalui unit bisnis terdekat.

Namun, sistem manual ini menimbulkan beberapa kendala, di antaranya proses pencairan uang yang lambat karena harus menunggu unit bisnis menyeterkan minyak ke unit utama terlebih dahulu sebelum dana dapat diteruskan ke nasabah. Hal ini berpotensi membuat masyarakat kurang termotivasi untuk berpartisipasi dalam penyeteran minyak jelantah karena waktu tunggu yang lama dan ketidakefisienan pencatatan. Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan metode Agile Kanban dalam pengembangan sistem informasi pencatatan dan pengelolaan Bank Jatah Indonesia. Dengan pendekatan kolaboratif, sistem ini diharapkan mampu mempercepat proses validasi dan pencairan saldo, meminimalkan kesalahan pencatatan, serta meningkatkan partisipasi masyarakat dalam program pengelolaan minyak jelantah.



Gambar 2.1 Bank Jatah Indonesia

2.1.2 *Sistem Informasi*

Sistem informasi merupakan serangkaian komponen yang bekerja bersama untuk mengumpulkan, mengelola, menyimpan, memproses, dan menyebarkan informasi yang diperlukan untuk mendukung pengambilan keputusan dalam suatu organisasi atau entitas. Salah satu komponen utama dalam sistem informasi adalah pemanfaatan teknologi berbasis perangkat keras (hardware) dan perangkat lunak (software), yang bekerja secara terintegrasi bersama prosedur dan sumber daya manusia untuk memastikan informasi yang dihasilkan akurat, relevan, dan tepat waktu bagi kebutuhan organisasi (Fahmi Adham, 2024).

Sistem informasi juga dapat dipahami sebagai kerangka kerja yang menggabungkan dua elemen penting, yaitu sistem dan informasi. Sistem memberikan kerangka kerja untuk mengelola aktivitas dengan lebih terstruktur, sementara informasi yang diperoleh dari pengolahan data memberikan nilai tambah dalam pengambilan keputusan yang lebih baik, sehingga kombinasi keduanya membentuk fondasi penting bagi berbagai solusi teknologi modern (Arifudin et al., 2024).

2.1.3 *Agile Software Development*

Metode agile software development atau biasa disebut dengan agile adalah proses iteratif dalam pembuatan sebuah perangkat lunak. Agile adalah sebuah

metode manajemen proyek yang menggunakan siklus pengembangan yang singkat, atau bisa disebut juga “sprint” untuk focus pada peningkatan berkelanjutan dalam pengembangan suatu produk atau layanan (Lutfiani et al., 2023). Dalam pengembangan agile, terdapat 7 macam model, diantaranya SCRUM, extreme programming (XP), kanban, Crystal Methodology, Dynamic Systems Development Method, Feature Driven Development, dan Lean Software Development.

Metode agile menjadi populer saat ini karena memberikan fleksibilitas kepada pengembang 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 (Yusril et al., 2021).

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

1. *“Fits just right” process*, dengan hanya menerapkan proses yang penting berdasarkan best practice.
2. *Continuous testing and validation*, dengan 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.



Gambar 2.2 Metode Agile Software Development

2.1.4 Kanban

Kanban merupakan metode manajemen aliran kerja berkelanjutan (*continuous flow*) yang berfokus pada penyampaian nilai secara dinamis tanpa terikat batasan siklus waktu kaku (*timebox*). Menurut Anderson & Carmichael, (2016), metode ini memvisualisasikan cara kerja sistem menggunakan papan visual (*Kanban board*) and kartu (*card*) yang bergerak melintasi kolom status tahapan kerja, seperti *To Do*, *In Progress*, dan *Done*. Melalui kombinasi visualisasi alur dan pembatasan pekerjaan dalam proses (*Work in Progress Limit*), Kanban bertujuan untuk mengendalikan kapasitas pengerjaan, menyepakati aturan kerja, serta menjaga kelancaran aliran tugas dalam tim.



Gambar 2.3 Prinsip Dasar Kanban

Kanban menerapkan empat prinsip dasar dalam manajemen perubahan sebagai berikut:

1. Memulai dari Apa yang Sudah Diketahui (Start with What You Know)
Metode Kanban diterapkan langsung pada alur kerja pengembangan sistem yang saat ini sedang berjalan tanpa merombak proses kerja tim pengembang secara instan.
2. Menyetujui untuk Melakukan Perubahan (Agree to Pursue Change)
Setiap anggota tim berkomitmen untuk melakukan perbaikan alur kerja dan pengembangan sistem secara bertahap serta berkelanjutan.
3. Menghormati Peran dan Tanggung Jawab saat Ini (Respect the Current Roles and Responsibilities)
Proses pengembangan tetap menghargai dan mempertahankan pembagian peran serta tanggung jawab anggota tim yang telah disepakati saat ini.
4. Mendorong Inisiatif Kepemimpinan (Encourage Acts of Leadership)
Setiap anggota tim diberikan kebebasan untuk mengambil inisiatif dan mengusulkan ide perbaikan demi kelancaran penyelesaian proyek.

Mengacu pada kerangka kerja Essential Upstream Kanban (Stayaert, 2018) serta Essential Kanban Condensed (Anderson & Carmichael, 2016), penerapan metode Kanban dalam pengembangan perangkat lunak memadukan fase Upstream Kanban dan System Kanban. Konsep Upstream Kanban berperan dalam fase penemuan serta penyaringan kebutuhan, sedangkan System Kanban digunakan untuk mengelola eksekusi pengembangan fitur.

1. Upstream Kanban
Pada tahap ini, fokus utamanya adalah menyaring kebutuhan secara teliti agar solusi yang dibuat benar-benar menjawab permasalahan pengguna (Stayaert, 2018). Berbagai ide dan permintaan dari pengguna dikumpulkan sebagai kandidat fitur, yang kemudian harus divalidasi kelayakannya sebelum mulai dikerjakan. Proses penyaringan ini bertujuan untuk menyeimbangkan banyaknya permintaan yang masuk dengan kemampuan kerja tim pengembang. Dalam implementasinya, tahapan penemuan (discovery) dan penyaringan kandidat fitur ini dijalankan melalui proses berikut:

a. Tahap Penangkapan (Capture)

Tahap mengumpulkan berbagai ide dan kebutuhan dari pemangku kepentingan yang awalnya masih belum terstruktur, untuk dijadikan sebagai opsi fitur awal.

b. Tahap Sistesis (Synthesis)

Tahap memformulasikan kebutuhan yang terkumpul menjadi rancangan alur proses bisnis usulan serta daftar opsi fitur baru.

c. Tahap Analisis (Analysis)

Tahap memodelkan spesifikasi teknis sistem sebelum diajukan ke Titik Komitmen.

2. Titik Komitmen (Commitment Point)

Titik penentu di mana sebuah kandidat fitur resmi disetujui untuk dikerjakan. Sebelum melewati titik ini, ide fitur masih bisa dibatalkan atau ditunda dengan risiko yang minim. Setelah melewatinya, status fitur berubah menjadi pekerjaan pasti (committed work) dan fokus tim beralih pada penyelesaian kode. Titik komitmen mencakup dua aktivitas:

a. Seleksi Opsi (Option Selection)

Proses evaluasi dan penentuan prioritas terhadap daftar opsi fitur untuk memilih mana yang paling bernilai dan layak untuk dikerjakan.

b. Komitmen (Commitment)

Proses penetapan jadwal dan persetujuan resmi bahwa tim akan berkomitmen penuh untuk mulai mengerjakan opsi fitur tersebut.

3. System Kanban

Tahap ini berfokus pada proses penulisan kode program menggunakan sistem tarik (pull system) untuk menjaga kelancaran alur kerja hingga fitur siap dirilis.

Pelaksanaan tahap ini menerapkan enam praktik utama meliputi:

a. Visualisasi (Visualize)

Memvisualisasikan alur kerja dan proses yang dilalui pekerjaan melalui papan Kanban dengan kolom status seperti To Do, In Progress, Review, dan Done.

b. Batasi Pekerjaan dalam Proses (Limit Work in Progress)

Membatasi jumlah tugas yang sedang dikerjakan secara bersamaan untuk mencegah penumpukan dan menjaga fokus tim.

c. Mengelola Aliran (Manage Flow)

Memantau pergerakan pekerjaan secara terus-menerus untuk memastikan proses pengembangan berjalan lancar tanpa hambatan teknis.

d. Buat Kebijakan Eksplisit (Make Policies Explicit)

Menetapkan aturan dan standar penyelesaian yang jelas untuk setiap tahapan pekerjaan.

e. Terapkan Umpan Balik (Implement Feedback Loops)

Melakukan evaluasi dan koordinasi untuk meninjau perkembangan serta menyelaraskan proses pengembangan.

f. Tingkatkan Secara Kolaboratif (Improve Collaboratively)

Melakukan tindakan perbaikan dan penyempurnaan alur kerja secara berkelanjutan berdasarkan hasil evaluasi.

2.1.4.1 Pengukuran Aliran Kerja (Flow Metrics)

Pengukuran aliran kerja dalam metode Kanban digunakan untuk memantau jalannya proses pengembangan sistem. Hasil pengukuran ini menjadi acuan untuk mengevaluasi kinerja aliran kerja (Anderson & Carmichael, 2016). Dua pengukuran utama dalam aliran kerja Kanban adalah sebagai berikut:

1. *Lead Time*

Total waktu yang dibutuhkan untuk menyelesaikan satu item pekerjaan, dihitung mulai dari saat pekerjaan tersebut dikomitmenkan pada *Commitment Point* hingga pekerjaan tersebut selesai sepenuhnya dan siap diserahkan pada *Delivery Point*. *Lead Time* mencerminkan seberapa cepat sistem dapat memenuhi permintaan pelanggan.

2. *Delivery Rate atau Throughput*

Jumlah item pekerjaan yang berhasil diselesaikan atau *completed* per satuan waktu. Dalam penelitian ini, *Throughput* diukur berdasarkan jumlah kartu pekerjaan yang mencapai status selesai (*Done*) untuk mencerminkan tingkat produktivitas tim pengembang dalam menghasilkan nilai.

2.1.4.2 *Click Up*

ClickUp merupakan platform produktivitas yang dirancang untuk menyatukan berbagai kebutuhan kerja tim dalam satu ekosistem, sehingga dapat mengurangi ketergantungan pada pemakaian banyak aplikasi yang terpisah. Sistem ini mengelola alur pekerjaan menggunakan struktur yang terorganisir mulai dari Workspace, Space, Folder, List, hingga Task. Selain berfokus pada manajemen tugas, ClickUp juga mendukung aktivitas kolaborasi melalui pemanfaatan fitur dokumen (Docs), media visual (Whiteboards), pelaporan kinerja (Dashboards), serta bantuan kecerdasan buatan (ClickUp Brain) yang ditujukan untuk meningkatkan efisiensi operasional perusahaan (ClickUp, 2026).

2.1.5 *Android*

Android merupakan sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak seperti smartphone dan tablet, dengan menyediakan platform terbuka bagi para pengembang untuk membangun aplikasi menggunakan bahasa pemrograman Java atau Kotlin (Kusniyati & Sitanggang, 2016). Android menyediakan berbagai komponen seperti Activity, Service, Broadcast Receiver, dan Content Provider yang berfungsi sebagai dasar dalam pengembangan aplikasi. Sistem operasi ini bersifat open-source sehingga dapat dimodifikasi dan dikembangkan sesuai kebutuhan aplikasi yang dibangun. Android juga dilengkapi dengan Android SDK (Software Development Kit) yang berisi tools, emulator, dan pustaka API untuk mempermudah proses pengembangan aplikasi.

2.1.6 *Kotlin*

Kotlin merupakan bahasa pemrograman modern yang menggabungkan paradigma berorientasi objek dan fungsional, dikompilasi menjadi Java bytecode sehingga dapat dijalankan di atas Java Virtual Machine (JVM) dan sepenuhnya interoperable dengan Java. Sejak Google mengadopsi Kotlin sebagai bahasa resmi pengembangan Android pada tahun 2017, para pengembang banyak beralih ke Kotlin karena fitur-fitur unggulannya yang tidak tersedia di Java, seperti extension functions, lambdas, smart casts, serta mekanisme null-safety yang mencegah terjadinya null-pointer exception. (Martinez & Gois Mateus, 2022)

Kotlin menyediakan fitur coroutine yang memungkinkan pengelolaan operasi asynchronous secara efisien. Dalam konteks pemanggilan REST API, coroutine mempermudah proses eksekusi jaringan tanpa memblokir thread utama, sehingga performa aplikasi menjadi lebih optimal. Ketika digabungkan dengan Retrofit, Kotlin mampu menangani request-response REST API secara lebih sederhana, aman, dan mudah dibaca (Nathasya, 2024).

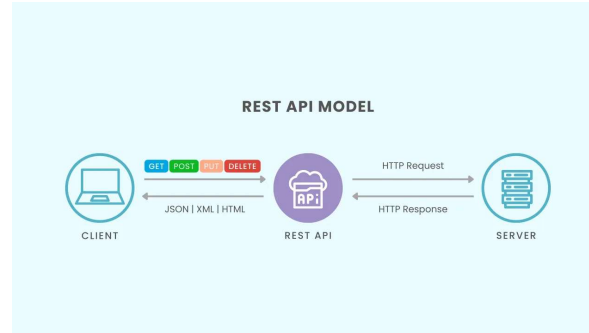
2.1.7 *Supabase*

Supabase merupakan platform backend-as-a-service (BaaS) berbasis open source yang dirancang untuk mempercepat proses pengembangan aplikasi tanpa perlu membangun sistem backend dari awal. Supabase dibangun di atas PostgreSQL dan menyediakan layanan yang komprehensif, meliputi autentikasi, database real-time, penyimpanan berkas, hingga fungsi serverless. Setiap proyek Supabase secara otomatis dilengkapi dengan API RESTful dan GraphQL yang dihasilkan langsung dari skema basis data, sehingga memungkinkan pengembangan aplikasi yang bersifat skalabel dan aman tanpa memerlukan pembangunan infrastruktur server dari awal (Supabase, 2024)

2.1.7.1 *Rest API*

REST API (REpresentational State Transfer Application Programming Interface) adalah gaya arsitektur web service yang memungkinkan aplikasi klien dan server berkomunikasi melalui protokol HTTP dengan pertukaran data menggunakan format standar seperti JSON atau XML. REST API memfasilitasi interoperabilitas antar sistem dengan cara yang ringan, fleksibel, dan mudah dipahami, sehingga komponen aplikasi yang berbeda seperti aplikasi mobile dan backend server dapat saling bertukar data tanpa tergantung pada platform atau bahasa pemrograman tertentu (Arianto & Susetyo, 2022).

REST API mendukung operasi dasar CRUD melalui metode HTTP seperti GET, POST, PUT, dan DELETE, serta menggunakan URL sebagai identifikasi sumber daya (resource) yang jelas. Dengan demikian, REST API memungkinkan pengembangan sistem yang modular, skalabel, dan mudah dipelihara karena klien dan server dapat dikembangkan secara terpisah dan berkomunikasi melalui antarmuka yang terstandarisasi (Herdiyatomoko, 2022).



Gambar 2.4 Arsitektur Rest API

2.1.7.2 *PostgreSQL*

PostgreSQL merupakan sistem manajemen basis data relasional open-source yang dirancang untuk menyediakan kestabilan, skalabilitas, serta kemampuan pengolahan data yang kompleks. PostgreSQL mendukung konsep objek-relasional, memungkinkan penggunaan tipe data kustom, fungsi buatan pengguna, serta ekstensi untuk menambah kapabilitas sistem sesuai kebutuhan pengembangan aplikasi modern (Praba & Safitri, 2020).

PostgreSQL menerapkan mekanisme Multi-Version Concurrency Control (MVCC) untuk menjamin konsistensi data saat banyak transaksi berlangsung secara paralel. Dengan MVCC, setiap transaksi memiliki versi datanya sendiri tanpa menghambat transaksi lain, sehingga kinerja sistem tetap optimal meskipun berada pada beban kerja tinggi (Ali et al., 2025).

Selain itu, PostgreSQL mendukung transaksi berstandar ACID, indexing yang beragam, replikasi, partisi tabel, serta optimasi query tingkat lanjut. Fitur-fitur ini menjadikan PostgreSQL cocok digunakan pada sistem informasi berskala besar, aplikasi enterprise, maupun layanan terdistribusi berbasis cloud (Uriawa et al., 2024). Fitur ini krusial bagi aplikasi yang menuntut tingkat konsistensi data yang tinggi serta kinerja transaksi yang dapat diandalkan.

2.1.8 *Use Case Diagram*

Teori Use Case Diagram menjelaskan bahwa diagram ini merupakan salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk memodelkan interaksi antara pengguna (aktor) dengan sistem yang akan dikembangkan. Use Case Diagram menggambarkan fungsionalitas sistem dari

sudut pandang pengguna, sehingga membantu memahami apa saja yang dapat dilakukan sistem tanpa harus menjelaskan bagaimana sistem tersebut bekerja secara internal (Wayahdi & Ruziq, 2023).

Secara umum, Use Case Diagram memiliki dua elemen utama, yaitu aktor dan use case. Aktor merupakan entitas eksternal yang berinteraksi dengan sistem, seperti pengguna, organisasi, atau perangkat lain. Sementara use case menggambarkan fungsi atau layanan yang disediakan oleh sistem untuk aktor tersebut. Hubungan antara aktor dan use case ditunjukkan melalui garis asosiasi yang menggambarkan interaksi atau komunikasi (Wayahdi & Ruziq, 2023).

Tujuan utama dari Use Case Diagram adalah untuk mengidentifikasi kebutuhan fungsional sistem, mempermudah komunikasi antara pengembang dan pengguna, serta menjadi dasar dalam perancangan dan pengujian sistem. Diagram ini juga membantu memastikan bahwa semua kebutuhan pengguna telah diakomodasi dalam desain sistem yang dibuat. Dengan demikian, Use Case Diagram berperan penting dalam tahap analisis dan perancangan sistem berbasis objek.

2.1.9 *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) merupakan salah satu model konseptual yang digunakan dalam perancangan basis data untuk menggambarkan hubungan antar entitas dalam suatu sistem. ERD membantu dalam memahami struktur data serta bagaimana data saling berhubungan di dalam sistem yang akan dikembangkan. Dengan kata lain, ERD berfungsi untuk memvisualisasikan logika dan keterkaitan antar tabel atau objek data sebelum diimplementasikan ke dalam basis data relasional (Afiifah et al., 2022).

Secara umum, ERD terdiri dari tiga komponen utama, yaitu entitas, atribut, dan relasi (relationship).

1. Entitas merepresentasikan objek nyata atau konsep yang memiliki data tersendiri, seperti User, Transaksi, atau Produk.
2. Atribut merupakan karakteristik atau informasi yang dimiliki oleh entitas, misalnya nama_user, tanggal_transaksi, atau harga_produk.
3. Relasi (relationship) menunjukkan hubungan antar entitas, misalnya satu User dapat melakukan banyak Transaksi (relasi one-to-many).

ERD juga dapat menunjukkan derajat hubungan (cardinality), seperti one-to-one (1:1), one-to-many (1:N), dan many-to-many (M:N), yang menjelaskan seberapa banyak entitas satu dapat berhubungan dengan entitas lainnya.

Tujuan utama dari pembuatan ERD adalah untuk memastikan struktur data yang konsisten, efisien, dan mudah dikelola, serta menjadi acuan utama dalam proses implementasi basis data fisik. Dengan menggunakan ERD, pengembang dapat merancang sistem penyimpanan data yang sesuai dengan kebutuhan fungsional sistem dan menghindari redundansi data.

2.1.10 *Black Box Testing*

Pengujian Black Box merupakan metode pengujian perangkat lunak secara fungsional yang bertujuan untuk mengetahui apakah sistem telah berfungsi dengan benar sesuai dengan kebutuhan pengguna. Pada metode ini, pengujian dilakukan dengan membuat berbagai kasus uji yang mencoba seluruh fungsi utama perangkat lunak berdasarkan spesifikasi yang telah ditetapkan. Fokus utama pengujian ini terletak pada keluaran (output) yang dihasilkan dari suatu masukan (input) tertentu tanpa memperhatikan bagaimana proses logika internal atau struktur kode di dalam sistem bekerja (Cholifah et al., 2018). Dengan demikian, pengujian ini menilai sejauh mana perangkat lunak memenuhi kebutuhan fungsional yang diharapkan oleh pengguna.

Selain itu, pengujian Black Box juga digunakan untuk mendeteksi berbagai jenis kesalahan seperti fungsi yang hilang, kesalahan pada struktur database, kesalahan pada antarmuka (interface), serta kesalahan dalam proses inisialisasi sistem (Permatasari et al., 2023). Dalam penerapannya, metode ini memiliki beberapa teknik pengujian, antara lain Boundary Value Analysis, Cause Effect Graphing, State Transition Testing, dan Equivalence Class Partitioning. Setiap teknik tersebut memiliki pendekatan tersendiri dalam menentukan data uji yang relevan sehingga hasil pengujian dapat lebih akurat dan mencakup berbagai kemungkinan kesalahan dalam sistem. Pendekatan ini menjadikan pengujian Black Box sebagai salah satu metode yang efektif dalam memastikan keandalan fungsi perangkat lunak sebelum digunakan secara luas.

2.1.11 *User Acceptance Testing (UAT)*

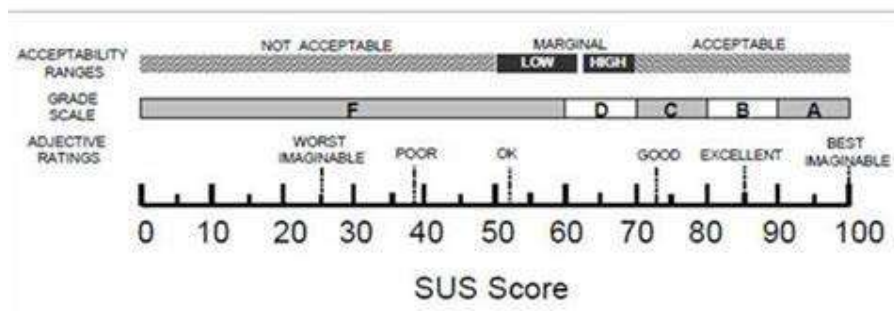
User Acceptance Testing (UAT) adalah tahapan pengujian akhir dalam pengembangan perangkat lunak yang dilakukan untuk memastikan bahwa sistem yang dibangun telah memenuhi kebutuhan dan harapan pengguna sebelum dinyatakan siap digunakan. Dalam UAT, pengguna akhir atau perwakilan pihak terkait secara langsung mengevaluasi sistem untuk memverifikasi bahwa fungsionalitas yang disediakan sesuai dengan kebutuhan bisnis dan skenario penggunaan sebenarnya. Pengujian ini biasanya dilakukan setelah pengujian fungsional dan sistem selesai, sehingga hasilnya dapat menjadi dasar keputusan penerimaan sistem oleh pengguna atau pemangku kepentingan. UAT membantu menjamin bahwa perangkat lunak layak diterima secara operasional serta sesuai dengan kebutuhan yang telah disepakati (Hikmah Abdillah & Najiyah, 2023).

2.1.12 *System Usability Scale (SUS)*

System Usability Scale (SUS) merupakan sebuah kuesioner yang dikembangkan oleh John Brooke pada tahun 1986. Kuesioner SUS terdiri dari sepuluh pernyataan yang memberikan wawasan mengenai kemudahan penggunaan suatu produk. Menurut Brooke (2013), tujuan dari kuesioner SUS adalah untuk mengumpulkan perspektif pengguna mengenai kegunaan sebuah produk dan kuesioner ini dapat diselesaikan dengan cepat, sehingga tidak memerlukan banyak waktu dari pengguna. Skala pada kuesioner SUS menggunakan skala Likert dengan 5 skala. Skala Likert digunakan untuk menilai sikap dan opini, di mana skala 1 menunjukkan ketidaksetujuan yang sangat tinggi, sedangkan skala 5 menunjukkan kesetujuan yang sangat tinggi.

Perhitungan kuesioner SUS dilakukan melalui beberapa tahapan(Nika et al., 2023):

1. Setiap pernyataan memiliki nilai antara 1 hingga 5.
2. Untuk pernyataan ganjil (1, 3, 5, 7, 9), kontribusi nilai diperoleh dengan mengurangi 1 dari skala yang diberikan.
3. Untuk pernyataan genap (2, 4, 6, 8, 10), kontribusi nilai diperoleh dengan mengurangi skala yang diberikan dari 5.
4. Jumlah total dari kontribusi nilai pernyataan ganjil dan genap kemudian dikalikan dengan 2,5 untuk mendapatkan skor SUS.
5. Skor akhir SUS berkisar antara 0 hingga 100.



Gambar 2.5 Skor Kusiner SUS

Tabel 2.1 Pertanyaan SUS

No	Pertanyaan	STS	TS	N	S	SS
		1	2	3	4	5
1.	Saya pikir saya akan sering menggunakan sistem ini.					
2.	Saya merasa sistem ini terlalu kompleks.					
3.	Saya merasa sistem ini mudah digunakan.					
4.	Saya pikir saya membutuhkan bantuan teknis untuk menggunakan sistem ini.					
5.	Fungsi dalam sistem ini terintegrasi dengan baik.					
6.	Ada terlalu banyak inkonsistensi dalam sistem ini.					
7.	Saya membayangkan kebanyakan orang akan belajar menggunakan sistem ini dengan cepat.					
8.	Saya merasa sistem ini sangat membebani saya untuk digunakan.					
9.	Saya merasa sangat percaya diri saat menggunakan sistem ini.					
10.	Saya perlu mempelajari banyak hal sebelum bisa mulai menggunakan sistem ini.					

2.2 Penelitian Terdahulu

Penelitian terdahulu berfungsi sebagai dasar acuan dan pembanding dalam pengembangan penelitian yang akan dilakukan. Melalui telaah terhadap penelitian-penelitian sebelumnya, dapat diperoleh gambaran umum mengenai metode, teknologi, dan pendekatan yang relevan dengan topik penelitian ini, yaitu pengembangan Aplikasi Mobile Bank Jatah Indonesia Menggunakan Metode Agile Kanban.

Penelitian oleh (Ramadhani et al., 2022) dengan judul Implementasi sistem informasi donasi minyak jelantah (SiMINAH) berbasis digital di Kota Dumai dengan pendekatan Lean Software Development mengembangkan sistem informasi terintegrasi untuk mengatasi permasalahan pengelolaan limbah minyak jelantah yang masih dilakukan secara manual. Hal ini berpotensi mencemari lingkungan. Hasil penelitian menunjukkan bahwa penerapan prinsip eliminasi waste dalam Lean Software Development mampu menghasilkan sistem yang efisien dan sesuai kebutuhan pengguna, dengan tingkat keberhasilan fungsional 100% dan nilai usability sebesar 87,3% (kategori sangat layak).

Penelitian oleh (Julianto, 2021) dengan judul “Rancang Bangun Aplikasi E-Jelantah berbasis Mobile Solusi Permasalahan Sampah”, bertujuan mempermudah proses pengumpulan dan penukaran minyak jelantah. Aplikasi ini dikembangkan menggunakan Software Development Lifecycle (SDLC) Waterfall dan Android Studio serta Firebase. Hasil penelitian menyebutkan Aplikasi dapat melakukan pencarian lokasi pengepul sampah dan dapat melakukan transaksi penukaran serta menghasilkan laporan transaksi berdasarkan periode transaksi.

Penelitian oleh (Wardhana et al., 2019) berjudul “Pengembangan Aplikasi Mobile Transaksi Bank Sampah Online Berbasis Android (Studi Kasus: Bank Sampah Malang)” dari Universitas Brawijaya, bertujuan untuk memberikan informasi kepada nasabah mengenai transaksi perbankan yang ada pada BSM. Dari sisi petugas penjemput yang dapat mengetahui lokasi penjemputan nasabah menggunakan layanan dari google maps API. Sistem dikembangkan dengan menggunakan Software Development Lifecycle (SDLC) dengan menggunakan bahasa pemrograman Java berbasis Android. Hasil penelitian menyebutkan aplikasi transaksi bank sampah online ini memiliki tingkat usability yang tergolong baik

dikarenakan hasil pengujian mendapati nilai persentase yang cukup tinggi yaitu mencapai 81,774%.

Penelitian oleh (Tri et al., 2021) dengan judul “Pengembangan Sistem Pengumpulan Minyak Jelantah Berbasis Web dan Embedded di Kota Malang” bertujuan meningkatkan efisiensi proses pengumpulan minyak jelantah melalui sistem terintegrasi antara web dan perangkat IoT. Sistem dikembangkan menggunakan metode Waterfall dengan teknologi NodeMCU ESP8266, sensor water level, MQTT, serta aplikasi web berbasis PHP dan JavaScript. Hasil penelitian menyebutkan bahwa sistem berhasil menampilkan ketinggian minyak secara realtime, proses pemesanan dan penjemputan dapat dilakukan melalui website, dan seluruh pengujian fungsional menunjukkan hasil yang valid.

Berdasarkan tinjauan pustaka, sebagian besar penelitian sebelumnya telah mengembangkan sistem informasi terkait pengelolaan minyak jelantah dan bank sampah melalui digitalisasi dasar seperti pencatatan transaksi, layanan donasi, penukaran minyak jelantah, serta monitoring sederhana baik pada platform web maupun mobile. Namun, pengembangan tersebut masih terbatas pada fungsi individual dan belum menggabungkan layanan secara terintegrasi lintas platform. Belum ada penelitian yang menghubungkan aplikasi mobile dengan sistem web admin dalam satu ekosistem yang memungkinkan sinkronisasi data, validasi, serta pengelolaan nasabah dan unit bisnis secara terpadu. Oleh karena itu, gap penelitian muncul pada kebutuhan sistem yang terhubung dan konsisten antar platform. Penelitian ini diusulkan untuk mengisi gap tersebut melalui pengembangan aplikasi mobile Bank Jatah Indonesia yang terintegrasi dengan web admin sebagai sistem pusat.

Tabel 2.2 Perbandingan Peneliti Terdahulu

Peneliti	Judul	Metodologi	Platform	Pengujian	Hasil
Ramadhani dkk. (2024)	Implementasi Sistem Informasi Donasi Minyak Jelantah (SiMINAH) Berbasis Digital di	Developme nt (LSD)	web dan mobile	black-box testing, User Acceptance Testing (UAT), dan usability testing.	Minyak jelantah melalui web dan mobile, dengan semua fitur valid (tingkat keberhasilan

Peneliti	Judul	Metodologi	Platform	Pengujian	Hasil
	Kota Dumai dengan Pendekatan Lean Software Development				100 persen) dan skor usability 87,3 persen, sehingga pencatatan lebih teratur dan pengelolaan donasi lebih efektif.
Muhammad Riswan Julianto (2021)	Rancang Bangun Aplikasi E-Jelantah berbasis Mobile Solusi Permasalahan Sampah	SDLC Waterfall	Aplikasi Mobile (Android Studio, Firebase)	Blackbox Testing	Hasil penelitian menyebutkan n Aplikasi dapat melakukan pencarian lokasi pengepul sampah dan dapat melakukan transaksi penukaran serta menghasilkan laporan transaksi berdasarkan periode transaksi.
Wisnhu Surya Wardhana, Herman Tolle, dan Agi Putra Kharisma (2019)	Pengembangan Aplikasi Mobile Transaksi Bank Sampah Online Berbasis Android (Studi Kasus: Bank Sampah Malang)	SDLC Waterfall	Aplikasi Mobile (Android, Google Maps API)	Blackbox Testing, Usability Testing	Hasil penelitian menyebutkan n aplikasi transaksi bank sampah online ini memiliki tingkat usability yang tergolong baik dikarenakan hasil pengujian

Peneliti	Judul	Metodologi	Platform	Pengujian	Hasil
					mendapati nilai persentase yang cukup tinggi yaitu mencapai 81,774%.
Adelia Tri Caeserine Pinem, Fajar Pradana, Adam Hendra Brata (2021)	Pengembangan Sistem Pengumpulan Minyak Jelantah Berbasis Web dan Embedded di Kota Malang	Scrum Kanban	framework codeigniter, PHP, CSS, dan javasript	Blackbox testing	Hasil penelitian menyebutkan hasil perancangan yang sebagai dasar dari proses implementasi sudah sama dengan hasil implementasi yang didapatkan dari proses implementasi. Hasil pengujian yang didapatkan dari 3 pengujian menghasilkan hasil yang valid.
Usulan Peneliti (2025)	Rancang Bangun Aplikasi Mobile Bank Jatah Indonesia Menggunakan Metode Agile Scrum (Studi Kasus: Bank Jatah Indonesia)	Agile Kanban	Aplikasi Mobile (Anroid Studio, Kotlin, Supabase)	Blackbox testing, User Acceptance Testing, Usability Testing	

