

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka Relevan

Penelitian oleh Schwaber & Sutherland (2020) berjudul “*The Scrum Guide*” menjadi referensi utama yang menjelaskan kerangka kerja *Scrum* secara komprehensif, termasuk tahapan *Sprint*, *Backlog*, *Sprint review*, dan *Retrospective*. Panduan ini menyediakan landasan metodologis dalam pengembangan perangkat lunak berbasis iterasi. Relevansinya terletak pada penerapannya sebagai acuan utama dalam pengembangan sistem manajemen aset berbasis web di PLN, sehingga alur kerja tim dapat terstruktur dan terukur.

Penelitian Geetha L. et al. (2025) berjudul “*Scrum Implementation in Software Projects*” membahas tantangan yang muncul dalam penerapan *Scrum*, seperti keterlibatan pemangku kepentingan yang rendah dan backlog yang kurang optimal. Solusi yang diusulkan adalah penerapan praktik *Sprint review* dan *Retrospective* yang konsisten. Penelitian ini relevan karena memberikan gambaran realistis tentang hambatan yang mungkin dihadapi dalam pengembangan sistem di PLN serta strategi penanganannya.

Dalam tabel 2.1 disajikan ringkasan dari sejumlah referensi jurnal ilmiah yang relevan, mencakup informasi mengenai judul, penulis, tahun terbit, inti dari isi jurnal, serta relevansi masing-masing referensi terhadap penelitian ini.

Tabel 2.1 Tabel Kajian Pustaka Relevan

No	Judul & Penulis	Tahun	Ringkasan Isi	Relevansi terhadap Penelitian
1	Pengembangan Sistem Informasi Manajemen Aset di Undip – Dzaky & Kurniawan	2023	Pengembangan sistem aset menggunakan metode <i>Scrum</i> dalam lingkungan	Studi kontekstual yang sangat serupa dengan proyek pengembangan sistem aset di PLN

			institusi pendidikan, menunjukkan efisiensi iteratif.	sebagai lembaga besar.
2	Asset Management System with Container-based Approach – Underhaug	2022	Sistem manajemen aset dengan kontainerisasi dan <i>Scrum</i> , menyoroti struktur tim kecil dan dokumentasi yang rapi.	Relevan dalam hal pendekatan terstruktur, fleksibilitas, dan dokumentasi saat menggunakan <i>Scrum</i> di tim kecil .
3	Asset Management with Structured <i>Agile</i> Approach – Kalstad et al.	2024	Membangun sistem manajemen aset yang kuat dengan pendekatan <i>Agile</i> , menekankan integrasi database yang solid.	Menguatkan pentingnya integrasi database dalam sistem aset yang dibangun agar data valid dan berkelanjutan.
4	Pengembangan Sistem Keuangan Masjid Berbasis Web & Android – Ardian et al.	2024	Mengembangkan aplikasi sosial menggunakan <i>Scrum</i> , membuktikan fleksibilitasnya dalam proyek non-korporat.	Memberikan referensi bahwa <i>Scrum</i> tidak terbatas di sektor industri, tapi juga relevan untuk institusi sosial atau publik seperti PLN.
5	Sistem Administrasi Kependudukan Berbasis Web dengan <i>Scrum</i> – Sasmito & Nishom	2020	Sistem layanan publik digital berbasis web dengan pendekatan <i>Scrum</i> , terbukti meningkatkan efisiensi pelayanan.	Sejalan dengan proyek digitalisasi aset di PLN untuk mempercepat proses manual menjadi otomatis.

6	<i>Agile Approach in IoT System Management – Guerrero-Ulloa et al.</i>	2023	<i>Scrum</i> dan <i>Agile</i> memberi keuntungan pada pengelolaan sistem berbasis IoT dan aset kompleks.	Menegaskan bahwa <i>Scrum</i> cocok untuk proyek sistem aset dengan variabel kompleks seperti di PLN.
7	Pengembangan Aplikasi SIMARGA – Muay et al.	2024	Menggabungkan <i>Scrum</i> dan Waterfall dalam pengembangan sistem manajemen aset desa, dengan fokus pada kolaborasi dan iterasi.	Relevan untuk memahami fleksibilitas metode <i>Scrum</i> dan peluang kombinasi metodologi untuk efisiensi pengembangan sistem.
8.	Implementasi MVC (Model View Controller) pada Perancangan Aplikasi Presensi Berbasis Web (Preparasi) - R. Nuraeni, S. Fitri, C. Riki	2023	Jurnal ini membahas penggunaan arsitektur MVC dalam pengembangan aplikasi presensi berbasis web. Di dalamnya terdapat landasan teori mengenai flowchart beserta simbol-simbolnya, serta penjelasan lengkap mengenai Unified Modeling Language (UML) termasuk	Menyediakan dasar teori flowchart dan simbol yang dapat digunakan untuk menjelaskan perancangan proses sistem manajemen aset Anda.

			diagram class dan diagram aktivitas.	
--	--	--	--------------------------------------	--

Dari berbagai kajian pustaka yang telah dibahas, dapat disimpulkan bahwa metode *Scrum* merupakan pendekatan yang efektif dalam pengembangan sistem informasi berbasis web, terutama untuk proyek yang memerlukan fleksibilitas dan iterasi cepat. Penerapan *Scrum* terbukti mendukung pengembangan sistem manajemen aset secara terstruktur, efisien, dan responsif terhadap perubahan kebutuhan pengguna. Selain itu, pengelolaan aset secara digital melalui sistem terintegrasi dinilai mampu meningkatkan akurasi, transparansi, dan efisiensi operasional. Oleh karena itu, penggunaan *Scrum* dalam pengembangan sistem aset di Divisi STI PT PLN (Persero) UID Jakarta Raya sangat relevan dan tepat sasaran.

2.2 Landasan Teori

1) Sistem

Sistem merupakan sekumpulan komponen yang saling berinteraksi untuk mencapai tujuan tertentu melalui proses yang terorganisasi. Menurut Dzaky & Kurniawan (2023), dalam konteks manajemen aset, sistem diartikan sebagai integrasi antara perangkat lunak, basis data, dan prosedur kerja yang bekerja secara terpadu untuk mengelola siklus hidup aset secara efektif. Pandangan ini sejalan dengan Underhaug (2022) yang menekankan bahwa sistem yang baik harus memiliki struktur jelas dan mekanisme komunikasi antarbagian yang efisien, misalnya melalui penggunaan pendekatan kontainerisasi. Kalstad et al. (2024) juga memperkuat gagasan ini dengan menunjukkan pentingnya integrasi yang solid antar komponen agar sistem dapat berjalan berkelanjutan dan menjaga konsistensi data.

2) Informasi

Sistem informasi merupakan kombinasi dari teknologi, manusia, dan prosedur yang dirancang untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi. Dzaky & Kurniawan (2023) mengimplementasikan sistem informasi manajemen aset berbasis *Scrum* yang terbukti efektif dalam

mengatur data aset secara iteratif. Muay et al. (2024) juga menekankan fleksibilitas sistem informasi dengan menggabungkan metodologi *Scrum* dan Waterfall, memungkinkan adanya kolaborasi intensif antar pengguna dan pengembang. Ardian et al. (2024) memberikan contoh penerapan sistem informasi pada pengelolaan keuangan masjid berbasis web dan Android, membuktikan bahwa konsep ini dapat digunakan baik dalam skala institusi besar maupun organisasi sosial.

3) Sistem Informasi

Sistem informasi merupakan kombinasi dari teknologi, manusia, dan prosedur yang dirancang untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi. Dzaky & Kurniawan (2023) mengimplementasikan sistem informasi manajemen aset berbasis *Scrum* yang terbukti efektif dalam mengatur data aset secara iteratif. Muay et al. (2024) juga menekankan fleksibilitas sistem informasi dengan menggabungkan metodologi *Scrum* dan Waterfall, memungkinkan adanya kolaborasi intensif antar pengguna dan pengembang. Ardian et al. (2024) memberikan contoh penerapan sistem informasi pada pengelolaan keuangan masjid berbasis web dan Android, membuktikan bahwa konsep ini dapat digunakan baik dalam skala institusi besar maupun organisasi sosial. Integrasi antara sistem informasi dan strategi pemasaran digital menunjukkan bahwa sistem tidak hanya berperan sebagai pengolah data, tetapi juga sebagai pendukung pengambilan keputusan (Fadli Prathama et al., 2021).

4) Website

Website adalah kumpulan halaman digital yang saling terhubung dan diakses melalui jaringan internet, berfungsi sebagai sarana penyampaian informasi maupun layanan. Ardian et al. (2024) memanfaatkan website sebagai platform utama untuk mengelola keuangan masjid, yang memudahkan pengguna dalam mengakses data dan laporan secara real-time. Sasmito & Nishom (2020) juga membuktikan bahwa pemanfaatan website dalam layanan administrasi kependudukan dapat meningkatkan efisiensi dan kualitas pelayanan publik.

Implementasi website sebagai sistem operasional juga ditunjukkan pada pengembangan aplikasi Penerimaan Siswa Baru berbasis web yang menyediakan fitur pendaftaran online, notifikasi email, dashboard monitoring, serta manajemen data calon siswa dan pembayaran, sehingga proses administrasi menjadi lebih efisien dan terintegrasi (Siswipraptini et al., 2020). R. Nuraeni et al. (2023) menambahkan bahwa website dapat dikembangkan dengan berbagai arsitektur seperti MVC (Model View Controller) yang memberikan struktur terorganisasi dalam membangun aplikasi berbasis web.

5) Manajemen Aset

Manajemen aset merupakan proses strategis yang mencakup pencatatan, pelacakan, pemeliharaan, dan pemanfaatan aset agar tetap produktif dan bernilai bagi organisasi. Dalam konteks organisasi besar seperti PT PLN (Persero), manajemen aset menjadi sangat penting karena skala aset yang besar dan keragaman jenisnya. Dengan pendekatan yang tepat, manajemen aset mampu meningkatkan efisiensi biaya, akurasi pengambilan keputusan, dan memperpanjang umur aset.

Kusumojati & Mediawati (2024), mengembangkan sistem informasi manajemen aset berbasis web di lingkungan pendidikan tinggi. Hasil penelitiannya menunjukkan bahwa sistem ini mampu mencatat inventarisasi alat laboratorium, memfasilitasi proses peminjaman aset, serta mendukung pelaporan yang akurat. Sistem ini menjawab tantangan umum dalam pengelolaan aset manual seperti duplikasi data, keterlambatan pencatatan, dan hilangnya dokumen aset. Sementara itu, Kesuma Dinata & Maula Rizki (2024), menyoroti pentingnya sistem pelacakan aset secara real-time melalui aplikasi web di sektor pemerintahan. Dalam studi kasusnya, sistem tersebut terbukti mampu mempercepat akses terhadap data aset dan mengurangi kesalahan pencatatan. Hasil ini menunjukkan urgensi digitalisasi manajemen aset, terutama bagi organisasi publik yang masih menggunakan metode manual.

Dari kedua studi tersebut, dapat disimpulkan bahwa penerapan sistem manajemen aset berbasis web meningkatkan efisiensi serta mengurangi potensi human error dalam pengelolaan data. Hal ini sangat relevan dengan konteks penelitian ini, di mana sistem informasi digital diperlukan untuk mendukung pengelolaan aset yang terstruktur, transparan, dan mudah diakses oleh seluruh pemangku kepentingan.

6) Sistem Informasi Berbasis Web.

Sistem informasi berbasis web (web-based information systems) adalah suatu sistem yang menyediakan layanan pengolahan, penyimpanan, dan distribusi data melalui jaringan internet, dengan antarmuka pengguna yang dapat diakses melalui peramban (browser). Sistem ini menggabungkan teknologi pemrograman web, basis data terintegrasi, dan sistem server untuk memungkinkan akses informasi secara real-time dan terdistribusi.

Menurut Hidayat et al. (2020), menekankan pentingnya integrasi antara front-end dan back-end dalam pengembangan sistem web. Sistem informasi manajemen untuk layanan *e-commerce* yang mereka bangun menunjukkan bagaimana pendekatan berbasis web mampu mengotomatisasi layanan pelanggan serta pengelolaan data internal secara efisien. Implementasi sistem informasi berbasis web pada lembaga pendidikan juga menunjukkan bahwa digitalisasi proses administrasi dan promosi mampu meningkatkan efektivitas pengelolaan organisasi serta memperluas jangkauan layanan secara daring (Purwanto et al., 2021). Pada karya tulis juga menunjukan bahwa digitalisasi administrasi dan penyampaian informasi secara daring mampu meningkatkan efektifitas pengelolaan data serta mempercepat distribusi informasi kepada yang membutuhkan (Yessy Fitriani et al., 2024).

Dari perspektif teori Emadi (2023), mengusulkan pendekatan pengembangan desain sistem informasi berbasis web yang memprioritaskan keterlibatan pengguna, kemudahan navigasi, serta siklus pengujian yang berkelanjutan. Hal ini sejalan dengan prinsip *agile development* yang juga diterapkan dalam penelitian ini melalui metode *Scrum*. Penerapan sistem berbasis web tidak hanya berfungsi sebagai media penyimpanan data, tetapi juga sebagai

sarana kolaboratif yang mendukung komunikasi internal serta distribusi informasi dalam organisasi (Dody et al., 2023).

Dengan mengacu pada teori-teori tersebut, sistem informasi manajemen aset berbasis web yang dikembangkan dalam penelitian ini dirancang agar mampu memberikan pengalaman pengguna yang optimal, memiliki struktur database yang rapi, serta dapat diakses secara real-time untuk memudahkan pemantauan aset. Selain itu, sistem ini juga diharapkan dapat mendukung proses transformasi digital perusahaan melalui teknologi web yang adaptif dan hemat biaya.

7) Kerangka Kerja *Scrum*

Scrum adalah kerangka kerja pengembangan perangkat lunak yang bersifat iteratif dan inkremental. *Scrum* merupakan bagian dari metodologi *Agile* yang menekankan pada fleksibilitas, keterlibatan pengguna, dan kecepatan pengiriman fitur secara bertahap. Framework ini dirancang agar tim dapat dengan cepat beradaptasi terhadap perubahan kebutuhan, sembari menjaga kualitas produk dan kolaborasi tim.

Menurut Schwaber & Sutherland (2020), dalam *The Scrum Guide*, *Scrum* memiliki beberapa komponen inti:

- a. *Product backlog*: daftar seluruh kebutuhan sistem.
- b. *Sprint*: iterasi waktu tetap (biasanya 1–4 minggu) untuk mengembangkan sebagian fungsional sistem.
- c. *Sprint Planning*: perencanaan tujuan dan tugas *sprint*.
- d. *Daily scrum*: pertemuan harian untuk memantau progres.
- e. *Sprint review* dan *Retrospective*: evaluasi hasil dan proses pengembangan.

Scrum telah terbukti efektif dalam mengelola proyek berskala kecil hingga menengah yang memiliki kebutuhan yang berubah secara dinamis. Penelitian oleh Al-Saqqa et al. (2020), menunjukkan bahwa metode ini mendorong efisiensi tim, mengurangi kegagalan proyek, serta meningkatkan kepuasan pengguna.

Hal serupa juga ditemukan oleh Reddy et al. (2021), yang menerapkan *Scrum* dalam proyek pengembangan sistem informasi dengan kebutuhan yang dinamis. Mereka menyimpulkan bahwa *Scrum* memperkuat struktur kerja tim, meningkatkan visibilitas terhadap progres proyek, dan memungkinkan adaptasi cepat terhadap perubahan kebutuhan.

8) Laravel

Laravel merupakan framework berbasis PHP yang banyak digunakan untuk membangun aplikasi web dengan pola arsitektur *Model-view-controller* (MVC). Framework ini menyediakan banyak fitur built-in seperti routing, validasi form, session management, hingga ORM (Eloquent) yang sangat membantu dalam mengelola basis data secara efisien. Menurut Ma'arif Lubis (2024), Laravel mendukung pengembangan sistem secara modular dan fleksibel, memudahkan developer untuk membangun fitur-fitur kompleks dalam waktu yang relatif singkat. Dalam proyek ini, Laravel digunakan untuk merancang backend sistem, mengelola alur data, serta mengatur komunikasi antara database dan antarmuka pengguna.

9) Tailwind CSS

Tailwind CSS adalah framework CSS utility-first yang memungkinkan pengembang untuk membangun tampilan antarmuka secara langsung melalui penambahan kelas-kelas HTML yang telah disediakan. Framework ini menekankan efisiensi dalam styling, di mana developer tidak perlu membuat file CSS secara terpisah. Dalam penelitian oleh Maulana et al. (2024), Tailwind terbukti mempercepat proses pembuatan antarmuka yang konsisten, responsif, dan user-friendly dalam platform *e-commerce* berbasis web. Penggunaannya dalam proyek ini bertujuan untuk menyusun layout form pendataan aset, tampilan riwayat aset, dan halaman dashboard dengan lebih cepat dan rapi.

10) MySQL

MySQL merupakan sistem manajemen basis data relasional yang digunakan secara luas dalam berbagai aplikasi web. Integrasi MySQL dengan Laravel melalui Eloquent ORM memberikan kemudahan dalam pembuatan dan pengelolaan tabel, relasi antar data, serta pelaksanaan *query* SQL. Salma Athallah et al. (2024), menjelaskan bahwa MySQL sangat efektif untuk pengelolaan data berskala menengah hingga besar dalam sistem berbasis Laravel, termasuk dalam sistem yang menangani data kesehatan dan pelacakan historis. Dalam konteks proyek ini, MySQL digunakan untuk menyimpan informasi aset, kategori, riwayat perbaikan, dan status ketersediaan aset.

11) GitHub

GitHub adalah platform kolaborasi pengembangan perangkat lunak yang berbasis sistem kontrol versi Git. Penggunaan GitHub memungkinkan pengembang untuk mengatur versi source code, melacak perubahan, dan berkolaborasi dalam tim secara efisien. Berdasarkan studi oleh Maulana et al. (2024), GitHub sangat membantu dalam pengembangan perangkat lunak berbasis tim karena menyediakan fitur *pull request*, *commit log*, serta pengelolaan *branch*. Dalam proyek ini, GitHub digunakan untuk menyimpan repositori kode dan mengelola proses iteratif dalam pengembangan fitur menggunakan metode *Scrum*.

12) Figma

Figma adalah tools desain antarmuka berbasis cloud yang memungkinkan tim untuk membuat prototipe dan wireframe secara kolaboratif. Figma dapat diakses melalui browser dan mendukung kerja tim secara real-time, sehingga mempermudah koordinasi antara desainer dan developer. Maulana et al. (2024), menunjukkan bahwa penggunaan Figma dalam perancangan antarmuka sangat mempercepat proses visualisasi desain dan perbaikan iteratif. Dalam proyek ini, Figma digunakan untuk merancang tampilan form input aset, halaman pencarian, serta interface QR code yang ramah pengguna.

13) Visual Studio Code

Visual Studio Code (VSCode) adalah text editor lintas platform yang dirancang untuk pengembangan aplikasi modern. Editor ini mendukung berbagai ekstensi yang memperkaya pengalaman pengembangan seperti auto-completion, debugging, dan integrasi Git. Ma'arif Lubis (2024), menjelaskan bahwa VSCode mempermudah proses coding dalam proyek web berbasis Laravel dan Tailwind karena ringan, mudah digunakan, dan sangat customizable. Dalam proyek ini, VSCode digunakan sebagai alat utama dalam menulis dan mengelola source code.

14) Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah suatu bahasa visual standar yang digunakan untuk memodelkan, mendokumentasikan, dan menyusun spesifikasi dari sistem perangkat lunak. UML tidak hanya membantu dalam menggambarkan struktur sistem yang kompleks, tetapi juga mendukung pemahaman yang lebih baik terhadap fungsi dan interaksi antar komponen sistem, baik oleh pengembang maupun pemangku kepentingan non-teknis.

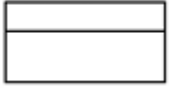
Menurut Nuraeni, Fitri, dan Riki (2023), penerapan UML dalam pengembangan sistem informasi berbasis MVC (*Model-view-controller*) memberikan kejelasan terhadap relasi antar entitas serta menggambarkan alur aktivitas sistem yang sedang dikembangkan. UML digunakan untuk mendeskripsikan proses bisnis dan logika sistem dalam bentuk visual, yang memudahkan kolaborasi dan analisis kebutuhan. Dalam penelitian mereka, penggunaan class diagram dan activity diagram digunakan untuk merepresentasikan struktur data serta alur proses presensi berbasis web. Hal ini juga menjadi acuan dalam penelitian ini untuk mengembangkan sistem manajemen aset.

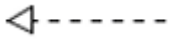
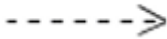
UML memiliki berbagai jenis diagram yang masing-masing memiliki fungsi tertentu adalah :

- a. Class Diagram: Menyediakan gambaran tentang struktur kelas dalam sistem, atribut yang dimiliki, serta relasi antara satu kelas dengan kelas lainnya.
- b. Activity Diagram: Menampilkan alur proses dan keputusan logika dalam sistem, sering digunakan untuk memodelkan business process atau workflow.
- c. Use Case Diagram : Menggambarkan interaksi antara *aktor* (pengguna atau sistem lain) dengan sistem yang akan dibangun. Diagram ini berfokus pada fungsi-fungsi (use case) yang dapat dijalankan oleh sistem dari sudut pandang pengguna.
- d. Sequence Diagram : Menggambarkan interaksi antar objek dalam urutan waktu tertentu. Fokusnya adalah pada pesan yang dikirim antar objek untuk menyelesaikan suatu proses.

Berikut adalah beberapa simbol class diagram.

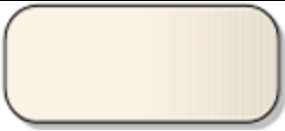
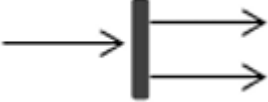
Tabel 2. 2 Tabel Simbol Class Diagram

Simbol	Nama Simbol	Keterangan
	Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
	Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	Class	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	Collaboration	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor

	Realization	Operasi yang benar-benar dilakukan oleh suatu objek.
	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya.
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Berikut adalah simbol activity diagram.

Tabel 2. 3 Tabel Simbol Activity Diagram

Simbol	Nama Simbol	Keterangan
	Status Awal/Initial	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal).
	Aktivitas/ Activity	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Asosiasi percabangan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	Penggabungan/ Join	Asosiasi penggabungan dimana lebih dari satu aktivitas lebih dari satu.
	Status Akhir/ Final	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status satu.

	Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
---	----------	--

Berikut ini adalah simbol Sequence Diagram.

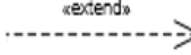
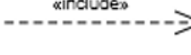
Tabel 2. 4 Tabel Simbol Sequence Diagram

Simbol	Nama Simbol	Keterangan
	LifeLine	Objek entity, antarmuka yang saling berinteraksi.
	Message	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktivitas yang sedang terjadi.
	Message	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktivitas yang sedang terjadi.

Berikut ini adalah beberapa simbol use case diagram dalam UML

Tabel 2. 5 Tabel Simbol Use Case Diagram

Simbol	Nama Simbol	Keterangan
	ACTOR	Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
	USE CASE	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling

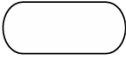
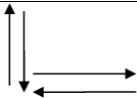
		bertukar pesan antar unit atau actor biasanya dinyatakan dengan menggunakan kata kerja di awal nama use case.
	ASOSIASI/ASSOCIATION	Komunikasi antara actor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor.
	EKSTENSI/EXTENSION	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan memiliki nama depan yang sama dengan use case yang di tambahkan.
	GENERALISASI/GENERALIZATION	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
	MENGGUNAKAN/INCLUDE	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsional atau sebagai syarat dijalankan use case ini.

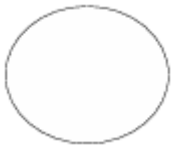
15) Flowchart

Flowchart adalah diagram alur yang digunakan untuk menggambarkan urutan langkah-langkah logis dalam suatu proses atau sistem secara visual. Flowchart berperan penting dalam memudahkan pemahaman terhadap sistem, baik oleh pengembang, pengguna, maupun pihak non-teknis lainnya. Flowchart umum digunakan pada tahap awal analisis sistem karena dapat menyederhanakan representasi proses yang kompleks menjadi visual yang lebih mudah dicerna.

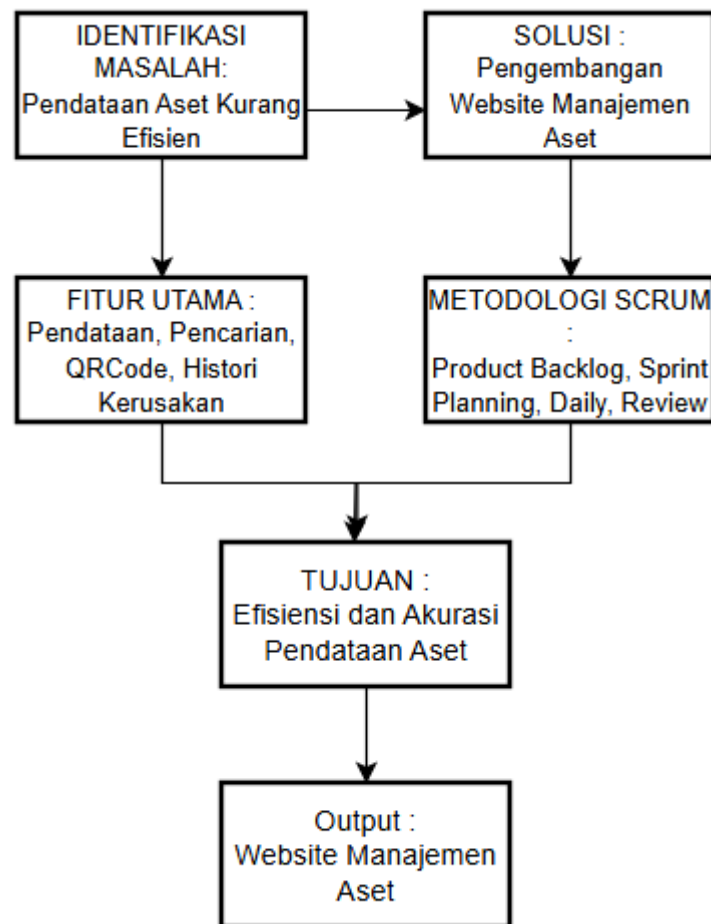
Menurut Nuraeni et al. (2023), flowchart adalah alat bantu yang sangat berguna dalam menggambarkan proses bisnis atau sistem yang akan dibangun, terutama pada tahap awal perancangan. Dalam studi mereka, flowchart digunakan untuk memodelkan proses presensi berbasis web secara logis dan runtut. Penggunaan flowchart dinilai membantu dalam menghindari inkonsistensi logika serta mempercepat komunikasi antara tim pengembang dan pengguna akhir.

Tabel 2. 6 Tabel Simbol Flowchart

Simbol	Nama Simbol	Keterangan
	TERMINAL	Digunakan untuk menggambarkan awal dan akhir dari suatu kegiatan.
	DECISION	Digunakan untuk menggambarkan proses pengujian suatu kondisi yang ada.
	PREPARATION	Digunakan untuk menggambarkan persiapan harga awal, dari proses yang akan dilakukan
	FLOW LINE	Digunakan untuk menggambarkan hubungan proses dari suatu proses ke proses lainnya.

	INPUT/OUTPUT	Digunakan untuk menggambarkan proses masukan data yang berupa pembicaraan data dan sekaligus proses keluaran yang berupa pencetakan data.
	SUBROUTINE	Digunakan untuk menggambarkan proses pemanggilan sub program dari main program (recursivitas).
	PROCESS	Digunakan untuk menggambarkan proses yang sedang dieksekusi.
	CONNECTOR	Digunakan sebagai penghubung antara suatu proses dengan proses lainnya yang ada dalam satu lembar halaman.
	PAGE CONNECTOR	Digunakan sebagai penghubung antara suatu proses dengan proses lainnya, tetapi berpindah halaman.

2.3 Kerangka Pemikiran



Gambar 2. 1 Diagram Kerangka Pemikiran

Gambar 2.1 menjelaskan tentang kerangka pemikiran yang merupakan landasan berpikir yang dirancang oleh peneliti untuk menggambarkan proses logis dari identifikasi permasalahan hingga penyusunan solusi yang ditawarkan. Dalam penelitian ini, pendekatan yang digunakan bersifat deduktif, yaitu berangkat dari permasalahan umum yang dihadapi organisasi, lalu dikerucutkan menjadi solusi sistematis melalui penerapan metode pengembangan perangkat lunak berbasis *Scrum*.

Permasalahan awal yang dihadapi oleh PT PLN (Persero) UID Jakarta Raya adalah sistem pengelolaan data aset yang masih dilakukan secara manual. Metode tersebut menimbulkan berbagai kendala, antara lain keterlambatan

pencatatan, kurangnya transparansi riwayat kerusakan, serta kesulitan dalam proses pelacakan aset secara efisien. Untuk menjawab permasalahan ini, dikembangkan sebuah sistem informasi manajemen aset berbasis website yang mampu mendigitalisasi proses pencatatan, pencarian, dan pelaporan aset.

Dalam proses pengembangannya, digunakan metode *Scrum* sebagai kerangka kerja. *Scrum* dipilih karena memiliki pendekatan iteratif dan fleksibel yang cocok untuk proyek dengan perubahan kebutuhan yang dinamis, seperti proyek pengembangan sistem dalam konteks industri. Tahapan yang dilakukan meliputi penyusunan *Product backlog*, perencanaan *Sprint*, pelaksanaan *Daily scrum*, dan diakhiri dengan sesi evaluasi melalui *Sprint review*.

Tujuan akhir dari sistem ini adalah untuk memberikan solusi nyata kepada PLN dalam mempercepat dan menyederhanakan proses pengelolaan aset, dengan hasil berupa sebuah sistem manajemen aset berbasis web yang siap digunakan dan mendukung kebutuhan operasional secara efisien.