

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era perkembangan transformasi digital, pemanfaatan teknologi berbasis web dalam dunia pendidikan menjadi semakin penting untuk mendukung berbagai aktivitas akademik. Penggunaan aplikasi berbasis web dinilai mampu meningkatkan efisiensi, kemudahan akses, serta membantu dalam pengelolaan data dan proses secara lebih terstruktur (Setiawan et al., 2025). Namun, dalam praktiknya, masih terdapat beberapa kegiatan akademik yang belum memanfaatkan teknologi secara optimal. Salah satu kegiatan akademik tersebut adalah *Acadion (Academic Integration)*, yaitu kegiatan yang di dalamnya terdapat berbagai rangkaian aktivitas, termasuk pelaksanaan *Capstone Project*. Kegiatan ini merupakan bagian dari program pengembangan talenta digital yang bertujuan membantu mahasiswa mengembangkan kompetensi sesuai kebutuhan industri serta mempersiapkan peserta menjadi *tech talent* di masa depan. Dalam pelaksanaannya, terdapat beberapa proses yang harus dikelola, seperti pembentukan kelompok, pengelolaan data peserta, pengumpulan worksheet, validasi kelompok dan worksheet oleh admin, serta pemberian umpan balik terhadap hasil pekerjaan mahasiswa.

Saat ini, pengelolaan *Capstone Project* masih dilakukan menggunakan beberapa media yang terpisah, seperti spreadsheet untuk pencatatan data dan email sebagai media pengiriman dokumen. Penggunaan spreadsheet memberikan kemudahan dalam pencatatan serta transparansi data, namun dalam pelaksanaan *Capstone Project* yang melibatkan banyak peserta, admin harus melakukan pembaruan data, pengecekan status kelompok, dan rekapitulasi informasi secara manual. Selain itu, data kelompok, hasil validasi, dan dokumen worksheet belum tersimpan dalam satu sistem sehingga proses pengelolaan masih bergantung pada beberapa media yang berbeda. Kondisi tersebut menyebabkan proses monitoring pengumpulan worksheet dan validasi masih dilakukan secara manual, meningkatkan risiko terjadinya ketidaksesuaian data akibat pembaruan manual,

serta menyulitkan admin dalam melakukan pelacakan riwayat aktivitas setiap kelompok.

Capstone Project merupakan program pembelajaran yang dirancang untuk mengintegrasikan pengetahuan dan keterampilan mahasiswa dalam menyelesaikan permasalahan nyata di lingkungan industri maupun masyarakat (Sugianto et al., 2025). Dalam pelaksanaannya, *Capstone Project* tidak hanya menuntut kemampuan teknis, tetapi juga kemampuan dalam perencanaan, kolaborasi, serta pelaporan hasil secara sistematis. Oleh karena itu, diperlukan suatu pengelolaan yang mendukung pelaksanaan setiap tahapan kegiatan sesuai dengan alur proses yang telah ditetapkan.

Berdasarkan permasalahan tersebut, diperlukan rancang bangun aplikasi Acadion berbasis web yang mampu mengintegrasikan proses pengelolaan *Capstone Project* dalam satu sistem. Aplikasi ini dirancang untuk mendukung pengelolaan data peserta, pembentukan kelompok, validasi kelompok dan worksheet, monitoring pengumpulan worksheet, serta penyampaian informasi kegiatan secara terpusat. Dengan adanya aplikasi Acadion, seluruh proses pengelolaan *Capstone Project* diharapkan dapat dilakukan melalui satu sistem berbasis web yang mengintegrasikan berbagai aktivitas administrasi dan pengelolaan data sesuai dengan kebutuhan proses bisnis yang telah diidentifikasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang, rumusan masalah dalam penelitian ini adalah “Bagaimana merancang dan membangun aplikasi Acadion berbasis web untuk mendukung pengelolaan *Capstone Project* secara terintegrasi?”.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

1. Sistem yang dikembangkan berupa Aplikasi Acadion berbasis web (Front-End dan Back-End) yang difokuskan pada pengelolaan *Capstone Project*.
2. Aplikasi ini dirancang untuk dua jenis pengguna yaitu admin dan student.
3. Sistem dikembangkan dan diuji dalam lingkungan *Local Server*.

1.4 Tujuan dan Manfaat

a. Tujuan Penelitian

Tujuan penelitian ini adalah merancang dan membangun Aplikasi Acadion berbasis web untuk pengelolaan *Capstone Project* yang mampu mengintegrasikan proses pengelolaan data, pembentukan kelompok, validasi kelompok dan *worksheet*, monitoring pengumpulan *worksheet*, serta pengelolaan *feedback*.

b. Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Bagi Mahasiswa: penelitian ini memberikan manfaat sebagai sarana untuk mengimplementasikan pengetahuan di bidang informatika, memperoleh pengalaman dalam proses perancangan dan pembangunan aplikasi, serta sebagai salah satu persyaratan untuk menyelesaikan studi.
2. Bagi Universitas: penelitian ini dapat menjadi referensi untuk penelitian sejenis serta menambah literatur terkait pengembangan aplikasi berbasis web dalam pengelolaan *Capstone Project*.

1.5 Metodologi Penelitian

Penelitian ini memanfaatkan dua pendekatan utama, yakni metode pengumpulan data dan metode pengembangan sistem, untuk menghasilkan keluaran yang sesuai dengan tujuan penelitian.

a. Pengumpulan Data

Metode pengumpulan data yang digunakan adalah sebagai berikut:

1. Observasi

Observasi merupakan metode pengumpulan data yang dilakukan melalui pengamatan langsung terhadap aktivitas yang berlangsung (Rozi et al., 2025). Pada penelitian ini, observasi dilakukan dengan mengamati proses pengelolaan *Capstone Project* yang melibatkan admin dan student sebagai peserta. Observasi difokuskan pada proses pembentukan kelompok, pengelolaan data peserta, pengumpulan *worksheet*, validasi kelompok dan *worksheet*, serta penyampaian informasi yang masih dilakukan menggunakan beberapa media terpisah, seperti *spreadsheet* dan

email. Hasil observasi digunakan untuk mengidentifikasi permasalahan serta kebutuhan sistem yang akan dikembangkan.

2. Wawancara

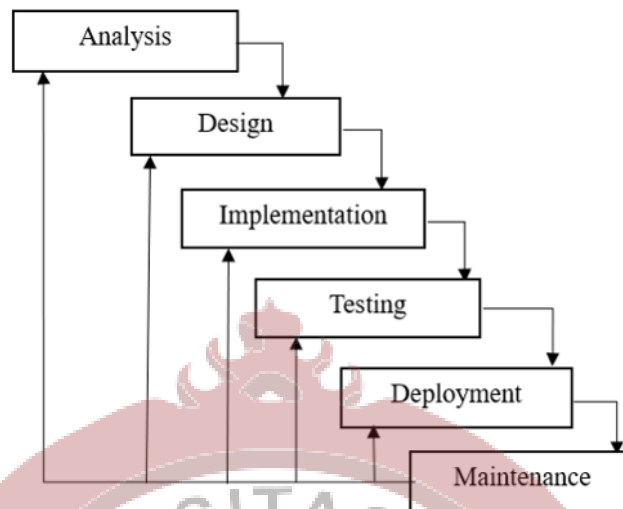
Wawancara merupakan teknik pengumpulan data yang dilakukan melalui interaksi langsung antara peneliti dan pihak terkait. (Rozi et al., 2025). Pada penelitian ini, wawancara dilakukan dengan Tim Program *Capstone Project 2025* yang bertanggung jawab dalam pengelolaan *Capstone Project*. Wawancara bertujuan untuk memperoleh informasi mengenai alur pengelolaan kegiatan, kendala yang dihadapi dalam proses administrasi, serta kebutuhan fungsional yang diperlukan sebagai dasar pengembangan Aplikasi Acadion berbasis web.

3. Studi Literatur

Studi pustaka dilakukan dengan menelaah berbagai sumber tertulis seperti buku, jurnal ilmiah, dan referensi lain yang berkaitan dengan pengembangan aplikasi berbasis web, *Capstone Project*, metode Waterfall, Unified Modeling Language (UML), dan Black Box Testing (Ibrohim et al., 2026). Studi pustaka bertujuan untuk memperoleh landasan teori, memahami konsep-konsep yang mendukung penelitian, serta menjadi acuan dalam proses analisis, perancangan, pengembangan, dan pengujian Aplikasi Acadion berbasis web.

b. Metode Pengembangan Sistem

Metode pengembangan yang digunakan dalam penelitian ini yaitu metode Waterfall. Metode Waterfall merupakan salah satu metode dalam System Development Life Cycle (SDLC) yang sering digunakan dalam pengembangan sistem informasi maupun perangkat lunak. Metode ini menggunakan pendekatan yang berurutan dan sistematis, di mana setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Proses dalam metode Waterfall dimulai dari tahap perencanaan hingga tahap pemeliharaan (*maintenance*) yang dilakukan secara terstruktur (Pratiwi et al., 2023). Waterfall pada penelitian ini ditampilkan pada Gambar 1.1.



Gambar 1. 1 Metode Waterfall (Pratiwi et al., 2023)

Dalam penelitian ini, metode Waterfall diterapkan melalui beberapa tahapan yang dilakukan secara sistematis sebagai berikut:

1. Analisis Kebutuhan Sistem

Pada tahap ini dilakukan identifikasi kebutuhan sistem berdasarkan permasalahan dalam pengelolaan *Capstone Project*. Analisis mencakup kebutuhan fungsional dan kebutuhan non-fungsional.

2. Desain

Tahap desain dilakukan untuk merancang arsitektur sistem, struktur database, serta tampilan antarmuka pengguna (*User Interface*). Perancangan sistem menggunakan pendekatan UML (*Unified Modeling Language*) yang meliputi *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan *Deployment Diagram*.

3. Implementation (Implementasi)

Tahap implementasi dilakukan dengan membangun sistem sesuai desain yang telah dibuat. Pengembangan *frontend* menggunakan ReactJS, sedangkan *backend* menggunakan NodeJS dan ExpressJS. Sementara itu, MySQL digunakan sebagai basis data untuk menyimpan dan mengelola data sistem.

4. Testing (Pengujian)

Tahap pengujian dilakukan untuk memastikan sistem berjalan sesuai dengan kebutuhan. Pengujian dilakukan menggunakan metode *Black Box Testing* untuk menguji fungsionalitas sistem serta Postman untuk pengujian API.

5. Deployment

Tahap *deployment* dilakukan dengan mengunggah sistem ke server agar dapat diakses oleh pengguna melalui jaringan internet. Proses ini mencakup konfigurasi *database*, *server backend*, serta domain aplikasi sehingga sistem dapat digunakan oleh admin dan user sesuai dengan hak akses masing-masing.

6. Maintenance (Pemeliharaan)

Tahap pemeliharaan dilakukan setelah sistem dibangun. Pada tahap ini dilakukan perbaikan bug, peningkatan performa, serta penyesuaian fitur apabila ditemukan kendala dalam penggunaan. Pemeliharaan bertujuan untuk memastikan sistem tetap stabil, aman, dan sesuai dengan kebutuhan pengguna.

1.6 Sistematika Penulisan

Untuk memudahkan pemahaman dan penyusunan laporan ini, sistematika penulisan disusun sebagai berikut.

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat penelitian, metode yang digunakan, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menguraikan tinjauan pustaka, kerangka berpikir, dan teori-teori yang mendukung penelitian, meliputi konsep pengelolaan Capstone Project, aplikasi berbasis web serta teknologi dan tools yang digunakan dalam pembangunan sistem, serta kerangka pemikiran yang menjadi dasar dalam perancangan dan pengembangan sistem.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas proses analisis kebutuhan sistem serta perancangan aplikasi Acadion berbasis web untuk pengelolaan *Capstone Project* secara terintegrasi. Perancangan dilakukan menggunakan UML yang mencakup *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini menjelaskan proses implementasi sistem berdasarkan desain yang telah dibuat, serta hasil pengujian sistem untuk memastikan seluruh fitur berjalan sesuai dengan kebutuhan yang telah ditentukan.

BAB V SIMPULAN DAN SARAN

Bab ini memuat simpulan dari hasil penelitian dan pengembangan sistem yang telah dilakukan, serta saran untuk pengembangan sistem di masa mendatang.



BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang dilakukan oleh (Nugraha et al., 2026) yang bertujuan untuk mengatasi permasalahan pengelolaan proyek secara manual yang menyebabkan keterlambatan dalam pelaporan. Penelitian ini menggunakan metode *Extreme Programming* (XP) yang terdiri dari empat tahapan. Hasil pengujian menggunakan *Black Box Testing* menunjukkan bahwa seluruh fungsi sistem berjalan dengan baik. Selain itu, pengujian *User Acceptance Testing* (UAT) terhadap sepuluh responden menunjukkan tingkat kepuasan sebesar 92% pada aspek kemudahan akses. Sistem yang dikembangkan mampu mendukung pengelolaan proyek, penugasan anggota, serta pelaporan secara real-time.

Penelitian oleh (Raihansyah, 2025) yang bertujuan untuk membangun sistem yang terstruktur dalam mendukung kebutuhan manajemen proyek di lingkungan perusahaan. Metode yang digunakan adalah Waterfall dengan pemodelan menggunakan BPMN dan UML yang terdiri dari 12 *use case* dan 15 activity diagram. Sistem dikembangkan menggunakan MySQL, CodeIgniter, dan Tailwind CSS. Hasil pengujian *Black Box Testing* terhadap 42 skenario menunjukkan tingkat keberhasilan 100%, serta sistem telah divalidasi oleh pihak terkait dan diimplementasikan pada *cloud*.

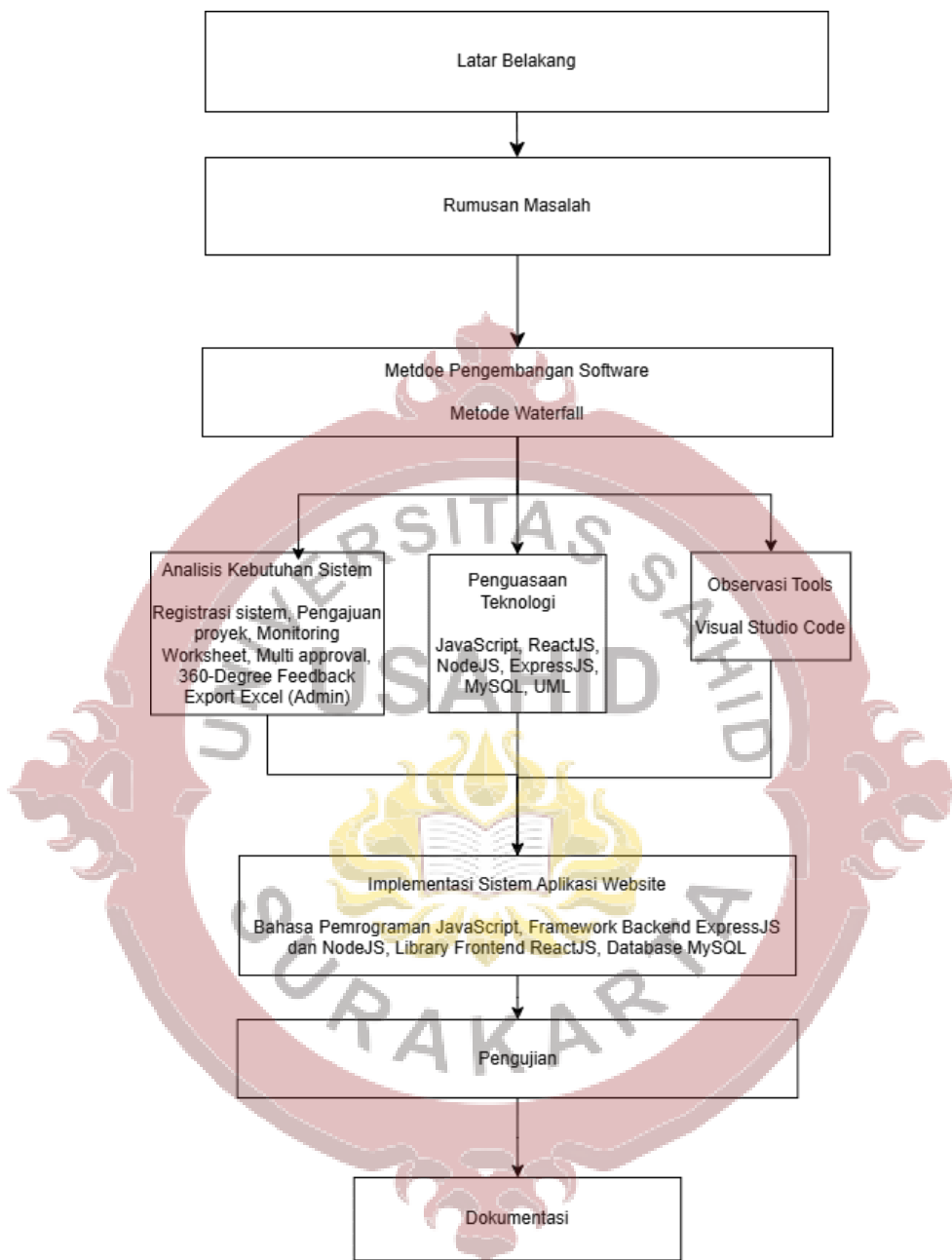
Penelitian oleh (Abdullah & Rusmawan, 2024) yang bertujuan meningkatkan efisiensi pengelolaan TA mahasiswa untuk koordinator program, dosen pembimbing, dan mahasiswa dalam memantau proses TA. Pengembangan mengikuti model Waterfall dengan metode pengumpulan data wawancara, studi lapangan, kajian literatur. Hasil pengujian *Black Box Testing* menunjukkan aplikasi berjalan lancar dan memenuhi kebutuhan pengguna untuk monitoring dan pengelolaan TA secara terstruktur.

Berbeda dengan penelitian-penelitian sebelumnya, penelitian ini berfokus pada rancang bangun Aplikasi Acadion berbasis web untuk pengelolaan *Capstone Project*. Aplikasi ini dilengkapi dengan fitur pengajuan proyek dan registrasi tim,

serta *monitoring* progres melalui pengisian *worksheet* berdasarkan jadwal yang telah ditentukan. Selain itu, sistem juga dilengkapi dengan mekanisme persetujuan (*approval*) dalam proses validasi tim dan validasi *worksheet*, fitur *Feedback* antar anggota, serta fasilitas export data dalam format Excel yang dapat diakses oleh admin.

2.2 Kerangka Pikiran

Kerangka pemikiran ini memberikan gambaran mengenai tahapan penelitian dalam merancang dan membangun Acadion sebagai sistem pengelolaan *Capstone Project* berbasis web. Kerangka ini menjelaskan alur dimulai dari identifikasi permasalahan dalam pengelolaan proyek yang masih belum terintegrasi, kemudian dilanjutkan dengan proses analisis kebutuhan sistem, perancangan menggunakan UML, hingga tahap implementasi dan pengujian sistem. Kerangka pemikiran ini diharapkan dapat memastikan setiap tahap pengembangan berjalan secara sistematis sehingga sistem yang dihasilkan mampu meningkatkan efektivitas, transparansi, dan kemudahan *monitoring Capstone Project*. Kerangka pemikiran dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Kerangka Pemikiran

Kerangka pemikiran ini menggambarkan langkah-langkah sistematis dalam mengembangkan *website*, berikut adalah penjelasan singkat setiap langkah:

1. Latar Belakang

Tahap awal penelitian menjelaskan bahwa pengelolaan kegiatan mata kuliah *Capstone Project* di Institusi XYZ masih dilakukan menggunakan spreadsheet dan surat elektronik (email). Kondisi tersebut menyebabkan proses pengelolaan

data, pemantauan perkembangan proyek, koordinasi antar pengguna, serta dokumentasi umpan balik belum terintegrasi dalam satu sistem. Oleh karena itu, diperlukan aplikasi berbasis web yang mampu mendukung pengelolaan kegiatan mata kuliah *Capstone Project* secara terstruktur dan terintegrasi.

2. Rumusan Masalah

Rumusan masalah difokuskan pada pertanyaan: “Bagaimana merancang dan membangun aplikasi Acadion berbasis web untuk mendukung pengelolaan *Capstone Project* secara terintegrasi?”

3. Metode Pengembangan Software

Pada tahap ini, pengembangan sistem dilakukan menggunakan metode SDLC model Waterfall, yang terdiri dari tahapan analisis kebutuhan, desain sistem, implementasi, pengujian, *deployment*, dan pemeliharaan.

4. Analisis Kebutuhan Sistem

Tahap ini dilakukan untuk mengidentifikasi kebutuhan fungsional dan non fungsional sistem. Kebutuhan fungsional meliputi registrasi tim, pengajuan proyek, monitoring *worksheet* terjadwal, fitur *Feedback*, serta *export* data ke Excel oleh admin. Kebutuhan non fungsional mencakup keamanan data, kemudahan penggunaan, dan kestabilan sistem.

5. Penguasaan Teknologi

Pada tahap ini dilakukan pemahaman terhadap teknologi yang digunakan dalam pengembangan sistem. Teknologi tersebut meliputi JavaScript sebagai bahasa pemrograman utama, ReactJS untuk pengembangan *frontend*, NodeJS dan ExpressJS untuk *backend*, serta MySQL sebagai sistem manajemen database. Selain itu, perancangan sistem dilakukan menggunakan UML untuk memodelkan alur dan struktur sistem sebelum diimplementasikan.

6. Observasi Tools

Tahap observasi tools dilakukan untuk menentukan dan memanfaatkan perangkat pendukung seperti Visual Studio Code yang digunakan *sebagai code editor* dalam proses pengkodean. Penggunaan tools ini bertujuan untuk mendukung proses pengembangan agar berjalan lebih terstruktur dan efisien.

7. Implementasi Sistem

Pada tahap implementasi, sistem dikembangkan dalam bentuk aplikasi berbasis web dengan menggunakan ReactJS sebagai *frontend*, NodeJS dan ExpressJS sebagai *backend*, serta MySQL sebagai basis data. Seluruh fitur yang telah dirancang pada tahap analisis diintegrasikan ke dalam sistem agar dapat mendukung proses pengelolaan *Capstone Project* secara menyeluruh dan terpusat.

8. Pengujian

Setelah sistem selesai dikembangkan, dilakukan pengujian untuk memastikan seluruh fungsi berjalan sesuai dengan kebutuhan yang telah ditentukan. Pengujian dilakukan menggunakan metode *blackbox* untuk menguji fungsionalitas sistem dan Postman Collection untuk memastikan API berjalan dengan baik.

9. Dokumentasi

Tahap ini dilakukan penyusunan dokumentasi yang mencakup seluruh proses penelitian, mulai dari analisis kebutuhan, perancangan, implementasi, hingga pengujian sistem, yang kemudian disusun dalam bentuk laporan sebagai hasil akhir penelitian.

2.3 Landasan Teori

2.3.1 Unified Modeling Language (UML)

UML (*Unified Modeling Language*) merupakan teknik pemodelan yang digunakan untuk menggambarkan sistem, khususnya dalam pengembangan perangkat lunak. UML juga dapat diartikan sebagai seperangkat aturan dan notasi yang digunakan untuk menspesifikasikan, memvisualisasikan, serta mendokumentasikan sistem berbasis objek.

UML menyediakan berbagai elemen grafis yang digunakan dalam proses perancangan dan pengembangan sistem berbasis *Object Oriented Analysis and Design* (OOAD), di mana sistem dipandang sebagai kumpulan objek yang saling berinteraksi. UML versi terbaru terdiri dari berbagai jenis diagram yang umumnya

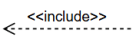
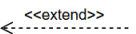
dikelompokkan menjadi dua kategori, yaitu *structure diagram* yang menggambarkan hubungan statis dalam sistem, dan *behaviour diagram* yang menggambarkan interaksi dinamis antar objek dalam sistem (Annisa Tri Hidayati et al., 2023).

UML memiliki berbagai jenis diagram yang digunakan untuk merepresentasikan permasalahan serta solusi dalam suatu sistem, di mana setiap diagram memiliki fungsi dan tujuan yang berbeda. Namun, dalam penelitian ini hanya empat diagram yang digunakan untuk menggambarkan rancangan sistem, yaitu sebagai berikut.

a. Use Case Diagram

Use case diagram merupakan salah satu alat yang digunakan dalam pengembangan sistem untuk menggambarkan hubungan antara aktor dengan sistem yang dibangun. Diagram ini biasanya dimanfaatkan untuk menjelaskan fungsi utama sistem secara umum serta menunjukkan ruang lingkup sistem yang dikembangkan. Selain itu, *use case diagram* juga digunakan untuk menggambarkan berbagai harapan atau kebutuhan pengguna terhadap sistem dari sudut pandang pihak yang berada di luar sistem. Dengan adanya diagram ini, perilaku sistem dapat dipahami berdasarkan interaksi yang dilakukan oleh pengguna (Wayahdi & Ruziq, 2023). Simbol Use Case Diagram dapat dilihat pada Tabel 2.1.

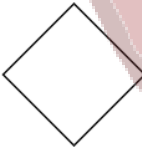
Tabel 2. 1 Use Case Diagram

Simbol	Keterangan
 Actor	Aktor: Merepresentasikan peran yang dapat berupa individu, sistem lain, maupun perangkat yang berinteraksi atau berkomunikasi dengan sistem yang dirancang.
	Use Case: Menggambarkan bentuk interaksi antara aktor dan sistem yang menunjukkan fungsi atau layanan yang disediakan sistem.
	Association: Menunjukkan hubungan atau keterkaitan antara aktor dengan use case yang menandakan adanya komunikasi atau interaksi di antara keduanya.
	Generalisasi: Menggambarkan hubungan pewarisan (inheritance) antar aktor, di mana aktor turunan memiliki karakteristik aktor induk dan dapat terlibat dalam use case yang sama.
 <<include>>	Include: Menunjukkan bahwa suatu use case sepenuhnya mencakup fungsionalitas dari use case lainnya sebagai bagian dari proses yang dijalankan.
 <<extend>>	Extend: Menunjukkan bahwa suatu use case dapat memperluas perilaku use case lain apabila kondisi tertentu terpenuhi.

b. Activity Diagram

Activity diagram merupakan salah satu jenis diagram dalam UML yang paling sering digunakan untuk menggambarkan alur proses dalam suatu sistem. Diagram ini digunakan untuk menunjukkan urutan aktivitas yang dijalankan serta menggambarkan aliran data atau komunikasi yang terjadi selama proses berlangsung. *Activity diagram* berfokus pada aliran kontrol (*control flow*) yang menghubungkan satu aktivitas dengan aktivitas lainnya sehingga proses yang terjadi dalam sistem dapat dipahami dengan lebih jelas. Diagram ini juga bersifat hierarkis, yang berarti suatu aktivitas dapat terdiri dari beberapa tindakan atau sub-aktivitas yang saling berkaitan dan membentuk suatu alur objek tertentu (Wayahdi & Ruziq, 2023). Simbol Activity Diagram bisa dilihat pada Tabel 2.2.

Tabel 2. 2 Activity Diagram

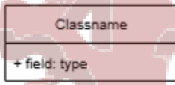
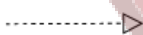
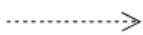
Simbol	Nama	Keterangan
	Status Awal (Initial State)	Simbol yang menandakan titik awal dimulainya suatu proses dalam diagram aktivitas. Setiap activity diagram memiliki satu status awal sebagai permulaan alur.
	Aktivitas (Activity)	Proses atau tindakan yang dilakukan dalam sistem. Aktivitas umumnya dituliskan dalam bentuk kata kerja karena menggambarkan suatu pekerjaan atau proses yang sedang berlangsung.
	Percabangan (Decision)	Simbol yang menunjukkan adanya pilihan atau kondisi tertentu yang menyebabkan alur proses bercabang ke lebih dari satu kemungkinan aktivitas.
	Penggabungan (Join)	Simbol yang digunakan untuk menyatukan kembali beberapa alur aktivitas yang sebelumnya terpisah menjadi satu alur proses.
	Status Akhir (Final State)	Simbol yang menunjukkan berakhirnya suatu proses dalam diagram aktivitas. Setiap activity diagram memiliki setidaknya satu status akhir sebagai penutup alur.
	Swimlane	Elemen yang digunakan untuk memisahkan tanggung jawab berdasarkan peran, bagian, atau unit organisasi yang terlibat dalam suatu proses, sehingga alur aktivitas menjadi lebih terstruktur dan jelas.

c. Class Diagram

Class diagram merupakan salah satu diagram dalam UML yang digunakan untuk menggambarkan struktur suatu sistem yang terdiri dari kelas, paket, dan objek beserta hubungan yang terjadi di antara komponen tersebut, seperti

association, *inheritance*, dan *containment*. Dalam konsep pemrograman berorientasi objek, *class* merupakan bentuk abstraksi yang mewakili karakteristik umum dari sekumpulan objek, baik dari segi struktur maupun perilakunya, sedangkan *object* merupakan instansiasi dari *class* yang dapat dibuat, dimodifikasi, maupun dihapus selama sistem dijalankan. Setiap objek memiliki kondisi yang mencerminkan nilai atribut serta keterkaitannya dengan objek lain dalam sistem. *Class diagram* juga berfungsi sebagai alat perencanaan sekaligus sarana komunikasi antar pengembang dalam proses pengembangan perangkat lunak karena mampu merepresentasikan struktur statis sistem dengan menampilkan kelas-kelas yang terdapat di dalam sistem beserta hubungan yang terbentuk di antaranya (Wayahdi & Ruziq, 2023). Simbol Class Diagram bisa dilihat pada Tabel 2.3.

Tabel 2. 3 Class Diagram

Simbol	Nama	Keterangan
	Generalization	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).
	Class	Sekumpulan objek yang memiliki atribut (data) dan operasi (metode) yang sama sebagai representasi dari suatu entitas dalam sistem.
	Collaboration	Gambaran interaksi atau rangkaian aksi antar objek dalam sistem yang bekerja sama untuk menghasilkan suatu keluaran atau tujuan tertentu bagi aktor.
	Realization	Hubungan yang menunjukkan bahwa suatu kelas atau objek mengimplementasikan operasi atau antarmuka tertentu secara nyata dalam sistem.
	Dependency	Hubungan ketergantungan di mana perubahan pada satu elemen independen dapat memengaruhi elemen lain yang bergantung padanya.
	Association	Hubungan struktural yang menggambarkan keterkaitan atau interaksi antara satu objek dengan objek lainnya dalam sistem.

d. Sequence Diagram

Sequence diagram merupakan salah satu diagram dalam UML yang digunakan untuk menggambarkan bagaimana objek dalam sistem saling berinteraksi dan bertukar pesan secara berurutan dalam rentang waktu tertentu. Diagram ini memperlihatkan bagaimana suatu aktivitas yang terdapat pada *use case* diterjemahkan ke dalam operasi yang dijalankan oleh objek pada class diagram. Penggunaan sequence diagram cukup luas karena sifatnya yang relatif mudah dipahami serta kemampuannya dalam menggambarkan sebagian perilaku sistem.

Fokus utama dari sequence diagram adalah menunjukkan interaksi antar objek sekaligus hubungan yang terjadi di antara objek tersebut. Dengan demikian, diagram ini dapat merepresentasikan interaksi antar komponen dalam sistem dan menjelaskan bagaimana objek saling bekerja sama untuk menjalankan suatu fungsi ketika sistem sedang berjalan (Wayahdi & Ruziq, 2023). Simbol Sequence Diagram bisa dilihat pada Tabel 2.4.

Tabel 2. 4 Sequence Diagram

Simbol	Nama	Keterangan
	Entity Class	Kelas yang merepresentasikan data atau entitas utama dalam sistem dan menjadi dasar dalam perancangan struktur basis data.
	Boundary Class	Kelas yang berfungsi sebagai penghubung antara sistem dengan aktor atau lingkungan luar, khususnya dalam menangani proses input dan output.
	Control Class	Kelas yang bertanggung jawab mengatur alur proses dan logika bisnis sistem serta mengoordinasikan interaksi antara entity class dan boundary class.
	Recursive Activation	Pesan yang dikirim oleh suatu objek kepada dirinya sendiri untuk menjalankan operasi tertentu secara berulang atau bertingkat.
	Activation	Representasi durasi eksekusi suatu operasi pada objek dalam sequence diagram, biasanya ditunjukkan dalam bentuk balok vertikal pada lifeline.
	Lifeline	Komponen yang menggambarkan keberadaan suatu objek selama proses interaksi berlangsung, biasanya ditampilkan dalam bentuk garis vertikal putus-putus pada sequence diagram.

e. Deployment Diagram

Deployment Diagram merupakan model yang digunakan untuk menggambarkan secara detail bagaimana komponen-komponen sistem ditempatkan dan disusun pada infrastruktur sistem (Aziz et al., 2024). *Deployment Diagram* digunakan untuk memvisualisasikan hubungan antara perangkat lunak dan perangkat keras. Secara khusus, diagram ini menggambarkan model fisik mengenai bagaimana komponen perangkat lunak (artefak) ditempatkan dan dijalankan pada komponen perangkat keras yang disebut node. *Deployment*

Diagram menjadi salah satu diagram penting pada tahap implementasi perangkat lunak dan biasanya dibuat sebelum proses coding dilakukan (Pulungan, 2023).

Tabel 2. 5 Deployment Diagram

Simbol	Deskripsi
	Package merupakan wadah atau pengelompokan yang terdiri dari satu atau lebih elemen dalam sistem, seperti kelas, diagram, atau komponen, yang disusun untuk menjaga struktur dan organisasi model agar lebih teratur dan mudah dipahami.
	Biasanya mencakup perangkat keras (<i>hardware</i>) maupun perangkat lunak eksternal (<i>software</i>) yang digunakan dalam sistem. Apabila di dalam suatu <i>node</i> terdapat komponen, maka komponen tersebut harus sesuai dengan elemen yang telah didefinisikan sebelumnya pada diagram agar rancangan sistem tetap konsisten.
	Ketergantungan antar <i>node</i> menunjukkan adanya hubungan penggunaan, di mana arah panah mengarah ke <i>node</i> yang menjadi tujuan atau yang digunakan oleh <i>node</i> lainnya.
	Component merupakan sebuah entitas <i>software</i> dalam sistem
	Relasi antar <i>node</i> merupakan hubungan atau koneksi yang menunjukkan adanya komunikasi atau keterkaitan antara satu <i>node</i> dengan <i>node</i> lainnya dalam arsitektur sistem.

2.3.2 Website

Website atau web merupakan sekumpulan halaman yang berisi informasi dalam bentuk digital, seperti teks, gambar, video, audio, dan animasi, yang dapat diakses melalui jaringan internet. Website ditampilkan melalui peramban (*browser*) seperti Mozilla Firefox, Google Chrome, dan *browser* lainnya, sehingga pengguna dapat memperoleh informasi secara mudah dan cepat. Website dapat dikembangkan menjadi aplikasi web yang mendukung proses bisnis melalui interaksi pengguna dan pengolahan data secara dinamis. (Arafat et al., 2022).

2.3.3 Tech Stack Front-End dan Back-end

Tech stack merupakan kumpulan teknologi dan *tools* yang digunakan secara bersamaan dalam proses pengembangan perangkat lunak atau aplikasi (Permono et al., 2024). Berikut merupakan *tech stack* yang digunakan:

1. Programming Language : Javascript

JavaScript merupakan bahasa pemrograman yang digunakan dalam pengembangan web untuk menciptakan tampilan yang interaktif, dinamis, dan responsif. Bahasa ini berjalan di sisi klien, yaitu dieksekusi melalui *browser*

pengguna, sehingga memungkinkan pengembang menambahkan berbagai fitur yang dapat meningkatkan pengalaman pengguna (Khoirurrizal et al., 2024).

2. Framework Front-end : React Js

ReactJS merupakan *library* front-end yang dikembangkan oleh Facebook dan banyak digunakan untuk mendukung pengembangan aplikasi web. ReactJS memiliki keunggulan seperti meningkatkan performa, kemudahan penggunaan (*simplicity*), serta skalabilitas. Selain itu, ReactJS memungkinkan pengembang membangun komponen antarmuka (UI) yang interaktif, bersifat *stateful*, dan dapat digunakan kembali (*reusable*) (Pramadipta, 2024). Dalam penelitian ini, ReactJS digunakan sebagai *framework* dari bahasa pemrograman JavaScript untuk mempermudah proses perancangan dan pembangunan aplikasi web.

3. Framework Backend : Express Js

Express.js merupakan *framework backend* berbasis Node.js yang ditulis menggunakan JavaScript dan digunakan untuk membangun aplikasi web secara lebih terstruktur. *Framework* ini mempermudah proses pengembangan melalui sintaks yang sederhana dan intuitif. Express.js juga menyediakan fitur *middleware* yang berfungsi untuk menangani berbagai proses seperti pengolahan data, manajemen error, dan autentikasi. Selain itu, sifatnya yang fleksibel memungkinkan pengembang untuk mengintegrasikan berbagai komponen sesuai kebutuhan. Dengan dukungan ekosistem Node.js yang luas, Express.js menjadi pilihan yang ringan, efisien, dan mudah dikembangkan untuk membangun aplikasi web modern (Nahjan et al., 2023).

4. Runtime Server : Node.js

Node.js merupakan lingkungan *runtime* yang bersifat *open-source* dan dapat dijalankan di berbagai platform seperti Linux, Windows, dan Mac. Node.js memungkinkan eksekusi bahasa pemrograman JavaScript di luar *browser*, sehingga dapat digunakan untuk pengembangan aplikasi di sisi server (Saputra & Prihandani, 2024).

5. Database : MySQL

MySQL merupakan sistem manajemen basis data relasional (RDBMS) yang bersifat *open-source* dan menggunakan SQL (Structured Query Language) untuk mengelola, mengakses, serta memanipulasi data. MySQL banyak digunakan dalam pengembangan aplikasi berbasis web dan menjadi salah satu database paling populer, terutama pada teknologi seperti PHP dan Apache. Kepopulerannya didukung oleh kinerja yang andal serta dapat digunakan secara gratis oleh para pengembang (Aziz et al., 2025).

6. *Version Control Code* : Github

GitHub merupakan platform berbasis cloud yang menyediakan layanan penyimpanan dan pengelolaan kode sumber menggunakan sistem kontrol versi Git. Selain digunakan oleh pengembang profesional, GitHub juga dapat dimanfaatkan oleh pemula sebagai media pembelajaran serta pengelolaan proyek berbasis web.

Penggunaan GitHub dalam pengembangan aplikasi web memberikan berbagai keunggulan, seperti kemudahan dalam pengelolaan file, pencatatan riwayat perubahan (*versioning*), serta dukungan kolaborasi antar pengembang. Selain itu, GitHub juga menyediakan layanan *GitHub Pages* yang memungkinkan pengguna untuk mempublikasikan website statis secara gratis (Jimmy et al., 2026).

2.3.4 Rancang Bangun

Rancang bangun merupakan proses perancangan dan pengembangan suatu sistem baru yang bertujuan untuk memberikan solusi terhadap permasalahan yang terjadi, sehingga sistem yang dihasilkan dapat berjalan secara efektif dan sesuai dengan kebutuhan pengguna (Jumardi, 2022). Dalam penelitian ini, rancang bangun mengacu pada pengembangan aplikasi Acadion berbasis web yang digunakan untuk mendukung pengelolaan *Capstone Project* secara terstruktur.

2.3.5 Aplikasi

Aplikasi merupakan perangkat lunak yang berfungsi untuk mengolah data melalui proses komputasi guna menghasilkan informasi yang dibutuhkan. Istilah aplikasi berasal dari bahasa Inggris *application* yang berarti penerapan atau

penggunaan. Secara umum, aplikasi diartikan sebagai program yang dirancang untuk membantu pengguna dalam menjalankan fungsi tertentu secara efektif sesuai dengan kebutuhan (Parjito et.al., 2022).

2.3.6 Capstone Project

Capstone Project merupakan metode pembelajaran berbasis proyek yang bertujuan untuk mengintegrasikan pengetahuan dan keterampilan mahasiswa dalam menyelesaikan permasalahan nyata di lingkungan industri maupun masyarakat. Dalam pelaksanaannya, mahasiswa bekerja secara tim melalui tahapan perencanaan, pengembangan, hingga evaluasi proyek.

Capstone Project juga berperan dalam meningkatkan kemampuan berpikir kritis, kolaborasi, serta manajemen proyek (Sugianto et al., 2025). Dalam penelitian ini, *Capstone Project* dikelola melalui aplikasi Acadion berbasis web untuk mendukung proses pengelolaan secara lebih terstruktur.

2.3.7 Pengelolaan

Pengelolaan merupakan proses mengatur dan mengembangkan suatu hal agar menjadi lebih baik serta memiliki nilai yang lebih tinggi dari kondisi sebelumnya. Pengelolaan dapat diartikan sebagai upaya untuk menjadikan suatu hal lebih baik serta sesuai dengan kebutuhan yang diharapkan (Hj. Saidah Hasbiyah, 2023).

2.3.8 Postman Collection

Postman Collection merupakan kumpulan request API yang terorganisir dalam satu koleksi, yang mencakup endpoint, metode HTTP (seperti GET, POST, PUT, DELETE), jenis otorisasi, parameter, header, body request, serta pengaturan lainnya. Setiap koleksi juga dapat dilengkapi dengan variabel dan dokumentasi pendukung. Postman Collection sangat bermanfaat dalam proses pengujian API, khususnya untuk *end-to-end testing*, pembuatan dokumentasi, serta mendukung kolaborasi dalam pengembangan aplikasi (Permono et al., 2024).

2.3.9 Black Box Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang berfokus pada pengujian fungsi-fungsi sistem tanpa memperhatikan atau mengetahui struktur maupun kode program di dalamnya. Pengujian dilakukan

dengan memverifikasi apakah keluaran yang dihasilkan telah sesuai dengan spesifikasi kebutuhan dan hasil yang diharapkan. Validitas pengujian pada metode ini diperoleh melalui kesesuaian antara spesifikasi sistem dengan hasil pengujian yang dihasilkan (Kartono et al., 2024).

