

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Untuk mendukung penelitian ini, penulis menggunakan beberapa tinjauan pustaka dari penelitian sebelumnya. Beberapa tinjauan tersebut adalah sebagai berikut:

Table 2.1 Daftar Literatur

No	Nama Peneliti	Judul Penelitian	Tahun
1	(Syaeful Anas Aklani & Jason Aaron Yang, 2022)	Pengembangan Dan Implementasi Tracer Study Di SMK Multi Studi High School Batam Menggunakan Framework Scrum	(2022)
2	(Ari Ardiyansyah & Budi Apriyanto, 2022)	Rancang Bangun Aplikasi Alumni dan Tracer Study Berbasis Web Menggunakan Metode (RAD) <i>Rapid Application Development</i> (Studi Kasus SMK Bhara Trikora 1 Jakarta)	(2022)
3	(Muhammad Yusuf & Ahmaddul Hadi, 2022)	Rancang Bangun Sistem E-Tracer Study Alumni SMKN 1 Lembah Melintang untuk Mengetahui Output Pendidikan Berbasis Web	(2022)
4	(Neha Dwivedi et al., 2022)	<i>A Comparative Study of Various Software Development Life Cycle (SDLC) Models</i>	(2022)
5	(Shelly Andari et al., 2021)	<i>Educational Management Graduates: A Tracer Study from Universitas Negeri Surabaya, Indonesia</i>	(2021)
6	(Ari Widayanti et al., 2021)	Tracer Study Alumni Program Studi Transportasi Untuk Mendukung Pengembangan Program Vokasi Universitas Negeri Surabaya	(2021)

No	Nama Peneliti	Judul Penelitian	Tahun
7	(Nando Hidayat & Habibah Nurfauziah, 2020)	Sistem Informasi Tracer Study Alumni Smk Negeri 1 Kota Bengkulu Berbasis Web	(2020)
8	(Prateek Rawat & Archana N. Mahajan, 2020)	<i>ReactJS: A Modern Web Development Framework</i>	(2020)
9	(Mega Isma Juwita et al., 2019)	Pengembangan Sistem Informasi Tracer Study Alumni Berbasis Web Menggunakan Metode RUP (Studi Kasus: SMA Suluh Jakarta Selatan)	(2019)
10	(Aminurlah Syam & Abdul Rachman Manga, 2017)	Sistem Tracer Study Alumni Fakultas Ilmu Komputer	(2017)

2.1.1 Literatur 1

Oleh Syaeful Anas Aklani, & Jason Aaron Yang (2022), dari Universitas Internasional Batam, dengan judul Pengembangan Dan Implementasi Tracer Study Di SMK Multi Studi High School Batam Menggunakan Framework Scrum. Dalam penelitian ini membahas bagaimana SMK Multi Studi High School Batam (SMK MHS) menjadi lebih mudah untuk mengumpulkan data dari alumni dengan mengembangkan dan menerapkan sistem informasi tracer study berbasis web. Menurut penelitian ini juga mereka mengklaim bahwa kurikulum sekolah yang kuno menghasilkan kurangnya kemampuan yang dapat dipekerjakan. Sistem pendidikan harus berubah dari era "*chalk and talk*" untuk menghasilkan siswa yang dapat dipekerjakan sesuai dengan kebutuhan industri. Metode evaluasi yang tepat

adalah satu-satunya cara untuk mengetahui apakah tujuan lembaga pendidikan tercapai atau tidak (Syaeful Anas Aklani & Jason Aaron Yang, 2022).

Untuk membuat dan menerapkan sistem informasi ini, penulis menggunakan metodologi Scrum. Sistem informasi berbasis web ini dibangun dan diimplementasikan dengan menggunakan Ruby on Rails sebagai backend, Devise sebagai solusi autentikasi, Bootstrap dengan template SB Admin 2 sebagai frontend, PostgreSQL sebagai sistem manajemen relational database (RDBMS), Git dan GitHub sebagai alat kontrol versi distribusi, dan Heroku sebagai platform hosting. Sistem informasi berbasis web ini memungkinkan alumni mengisi status secara mandiri, dan manajemen sekolah dapat melihat dan menilai data yang dikumpulkan (Syaeful Anas Aklani & Jason Aaron Yang, 2022).

Penelitian ini bertujuan untuk memberikan kontribusi positif dalam meningkatkan kualitas pendidikan di SMK Multi Studi High School melalui penerapan sistem informasi Tracer Study yang efektif dan efisien, mempermudah komunikasi dengan alumni untuk menjamin relevansi sistem pendidikan dengan kebutuhan industri, dan menyediakan data yang berguna bagi SMK MHS untuk melakukan perbaikan dan peningkatan kualitas pendidikan. Mereka juga dapat menggunakan informasi yang disajikan dalam penelitian ini untuk membangun sistem informasi tracer study yang sesuai dengan kebutuhan lembaga pendidikan vokasi.

2.1.2 Literatur 2

Oleh Ari Ardiyansyah, & Budi Apriyanto (2022), dari Universitas Pamulang, dengan judul Rancang Bangun Aplikasi Alumni dan Tracer Study Berbasis Web Menggunakan Metode (RAD) Rapid Application Development (Studi Kasus SMK Bhara Trikora 1 Jakarta). Dalam penelitian ini pengembangan aplikasi pengelolaan data alumni berbasis web untuk SMK Bhara Trikora 1 Jakarta. Tujuan dari aplikasi ini adalah untuk membantu pihak sekolah mengelola data alumni secara efektif, rapi, dan terstruktur. Metode pengembangan aplikasi cepat (RAD) adalah metode pengembangan aplikasi perangkat lunak yang menggunakan teknik untuk mempercepat proses pengembangan aplikasi. Beberapa teknik yang digunakan RAD untuk mencapai tujuan ini termasuk prototipasi, keterlibatan pemakai dalam pengembangan iteratif, dan alat prototipasi cepat.

Penelitian ini dilakukan menggunakan metodologi kualitatif deskriptif dengan referensi jurnal sebagai pendukung. Observasi, wawancara, dan analisis sistem saat ini digunakan untuk mengumpulkan data. Studi ini mengembangkan sebuah aplikasi berbasis web untuk mengelola data alumni yang memiliki laporan, informasi, kontak, galeri, akun pengguna, dan pengelolaan data alumni. Diharapkan aplikasi ini akan membantu manajemen data alumni SMK Bhara Trikora 1 Jakarta. Aplikasi ini memiliki kelebihan dalam efisiensi, kemudahan penggunaan, dan kemampuan untuk mengikuti jejak alumni. Namun, mereka juga memiliki kekurangan, seperti biaya pengembangan, instruksi pengguna, dan keamanan data yang diperlukan.

Penelitian ini memberikan wawasan terhadap pembaca yang mana dengan hasil penelitian ini secara keseluruhan membantu meningkatkan kualitas

pendidikan dengan memastikan program studi relevan dengan kebutuhan dunia kerja, memperkuat hubungan antara sekolah dan dunia kerja, dan membantu alumni sukses dalam karir mereka.

2.1.3 Literatur 3

Oleh Muhammad Yusuf, & Ahmad Hadi (2022), dari Teknik Informatika Fakultas Teknik, Universitas Negeri Padang, dengan judul Rancang Bangun Sistem E-Tracer Study Alumni SMKN 1 Lembah Melintang untuk Mengetahui Output Pendidikan Berbasis Web. Studi ini mempelajari bagaimana membuat sistem informasi e-tracer studi alumni yang berbasis web dan berbasis ponsel untuk SMKN 1 Lembah Melintang. Tujuan dari sistem ini adalah untuk membantu sekolah mengelola data alumni, melacak jejak mereka, dan memberikan informasi tentang pekerjaan mereka setelah lulus. Untuk memastikan pengembangan yang terstruktur dalam pengembangan perangkat lunak, penelitian ini menerapkan metode waterfall.

Penting bagi penelitian ini adalah menawarkan solusi bagi sekolah yang menghadapi masalah dalam mengelola data alumni secara manual, solusi ini meningkatkan efisiensi dan aksesibilitas. Penelitian ini membantu pembaca memahami bagaimana membuat sistem informasi e-tracer studi alumni yang dapat disesuaikan dengan kebutuhan institusi mereka sendiri. Ini juga membantu mereka memahami peran pengelolaan data alumni dalam proses evaluasi pendidikan dan hubungan dengan alumni.

2.1.4 Literatur 4

Oleh Neha Dwivedi, Devesh Katiyar, & Gaurav Goel (2022), dari *Department of Computer Science, Faculty of Computer & Information Technology*

Dr. Shakuntala Mishra National Rehabilitation University, Lucknow, India dengan judul *A Comparative Study of Various Software Development Life Cycle (SDLC) Models*. Penulis meneliti Siklus Hidup Pengembangan Perangkat Lunak, yang terdiri dari tujuh tahap: perencanaan, pengumpulan dan analisis kebutuhan, perancangan, pengkodean, pengujian, implementasi, dan pemeliharaan. yang secara keseluruhan merupakan persyaratan yang harus dipenuhi saat mengembangkan aplikasi apa pun.

Selain itu, penulis melakukan perbandingan model SDLC, yang biasanya digunakan dalam pengembangan proyek perangkat lunak. Ada model V-Model, Model Spiral, Model Air Terjun, dan Model Inkremental. Yang keempat memiliki keunggulan. Selain itu, penelitian ini membantu pembaca memperluas pengetahuan mereka tentang apa saja tahapan SDLC dan jenis model atau metode SDLC yang dapat diterapkan sesuai dengan skala proyek yang dikerjakan.

2.1.5 Literatur 5

Oleh Shelly Andari, Aditya Chandra Setiawan, Windasari, & Ainur Rifqi. (2021), dari jurusan Departemen Manajemen Pendidikan Universitas Negeri Surabaya, dengan judul *Educational Management Graduates: A Tracer Study from Universitas Negeri Surabaya, Indonesia*. Dalam penelitian ini menceritakan tentang studi tracer vokasi dan kejuruan di Universitas Negeri Surabaya, Indonesia. Studi ini menggunakan survei untuk mengumpulkan data dari lulusan vokasi dari program manajemen pendidikan universitas. Informasi yang dikumpulkan mencakup status pekerjaan, keberhasilan dalam mendapatkan pekerjaan, dan kemampuan lulusan untuk menerapkan pengetahuan dan keterampilan yang mereka pelajari.

Hasil penelitian menunjukkan bahwa lulusan vokasi dari Universitas Negeri Surabaya memiliki kinerja yang cukup baik. Sekitar 80% lulusan yang berhasil mendapatkan pekerjaan, sedangkan hanya 20% yang belum mendapatkan pekerjaan. Ini menunjukkan bahwa lulusan vokasi dari Universitas Negeri Surabaya memiliki kemampuan yang kuat untuk bekerja di industri apa pun.

Penelitian ini memberikan pengetahuan tentang kinerja lulusan vokasi di Universitas Negeri Surabaya, yang dapat digunakan sebagai referensi untuk meningkatkan kinerja vokasi dan kejuruan di universitas lainnya. Pembaca juga dapat menggunakan informasi dari literatur ini untuk mengembangkan strategi untuk memperbaiki kinerja vokasi dan kejuruan di universitas tersebut.

2.1.6 Literatur 6

Oleh Ari Widayanti, Fitri R. Widayanti, Anita Susanti, & Amanda R. Pattisinai (2021), dari Universitas Negeri Surabaya. Dengan judul Tracer Study Alumni Program Studi Transportasi Untuk Mendukung Pengembangan Program Vokasi Universitas Negeri Surabaya. Dalam penelitian ini mempelajari Tracer Study Alumni Program Studi Transportasi untuk mendukung pengembangan program vokasi di Universitas Negeri Surabaya. Tujuan dari penelitian ini adalah untuk mengetahui hasil pendidikan, seperti penilaian diri tentang penguasaan dan pemerolehan kompetensi, dan evaluasi proses pembelajaran dan kontribusi pendidikan tinggi terhadap pemerolehan kompetensi. Penelitian ini juga bertujuan untuk mengumpulkan data sosiobiografis alumni tentang input pendidikan.

Alumni disurvei melalui metode survei, dengan 58% perempuan dan 42% laki-laki. Tahun 2011, 2013, dan 2014 adalah tahun masuk kuliah terbanyak,

dengan 90,4% alumni sudah bekerja dan 9,6% belum. Sebagian besar alumni mendapat pekerjaan kurang dari 3 bulan, dan 25% sudah bekerja sebelum lulus. Meskipun demikian, 51,9% alumni menyatakan bahwa pekerjaan mereka tidak sesuai dengan bidang studi mereka.

Hasil dari penelitian tracer ini menunjukkan bahwa informasi yang dikumpulkan dapat bermanfaat untuk evaluasi dan penyempurnaan kualitas pendidikan tinggi serta memberikan wawasan tentang hubungan antara pendidikan tinggi dan dunia kerja. Studi ini juga berguna untuk menilai relevansi pendidikan tinggi dan memberikan informasi kepada pihak yang bertanggung jawab. Dibutuhkan upaya untuk membuat bidang pekerjaan alumni lebih sesuai dengan bidang studi mereka.

2.1.7 Literatur 7

Oleh Nando Hidayat, & Habibah Nurfauziah (2020), dari STMIK Muhammadiyah Jakarta, dengan judul Sistem Informasi Tracer Study Alumni Smk Negeri 1 Kota Bengkulu. Dalam penelitian ini Tracer Study adalah inisiatif sekolah untuk mengumpulkan profil status alumni terbaru. Alumni memiliki akses yang lebih baik ke lowongan kerja karena sekolah menyebarkan pencari kerja dan informasinya. Oleh karena itu, sekolah memerlukan suatu sistem yang dapat mengkomunikasikan korespondensi jabatan dan mencatat status alumni saat ini.

Dengan mengumpulkan data, penulis membuat sistem informasi berbasis web untuk studi lanjut alumni SMK Negeri 1 Kota Bengkulu dengan menggunakan wawancara, kuesioner, dan penelitian literatur. Bahasa pemrograman MySQL kompatibel dengan sistem ini. Karena sistem yang dibuat dengan baik, metode

waterfall digunakan dalam pengembangan perangkat lunak. Ini disebabkan oleh fakta bahwa prosesnya dilakukan secara bertahap, yang menghasilkan pengolahan yang konsisten.

Penelitian ini bertujuan untuk membantu pelaksana pendataan alumni secara efektif dan efisien dan juga untuk membantu alumni yang belum mendapatkan pekerjaan mencari pekerjaan. Dalam dokumen penelitian ini, program aplikasi web Tracer Study adalah metode yang lebih mudah dan cepat untuk mengumpulkan data alumni. karena lulusan dapat mengakses website dan mengisi data mereka. Program ini menggunakan formulir kuesioner yang disediakan untuk meminta alumni mengisi data pribadi terbaru mereka. Ini berbeda dengan pendekatan konvensional di mana sekolah meminta alumni untuk mengunjungi sekolah secara individu untuk mengisi informasi pribadi mereka.

2.1.8 Literatur 8

Oleh Prateek Rawat, Archana N. Mahajan (2020), dari *Student, Department of Computer Science and Engineering, Institute of Technology and Management Universe, Vadodara, Gujarat*, dengan judul *ReactJS: A Modern Web Development Framework*. Penelitian ini menjelaskan tentang framework ReactJS yang digunakan untuk membuat antarmuka pengguna (UI) aplikasi *single page* dengan baik. Pengembang dapat membuat aplikasi web besar yang dinamis tanpa memuat ulang halaman dengan menggunakan fitur ReactJS, seperti komponen yang dapat digunakan kembali dan virtual DOM. Selain itu, komunitas pengembang sangat menyukai filosofi ReactJS karena meningkatkan fokus programmer pada logika bisnis dan masalah yang lebih besar.

Pada penelitian ini memberikan pemahaman yang mendalam tentang fitur ReactJS dan bagaimana fitur tersebut dapat digunakan untuk mengembangkan aplikasi. Ini membantu pembaca dalam mengembangkan aplikasi web yang efisien dan berkualitas tinggi.

2.1.9 Literatur 9

Oleh Mega Isma Juwita, Satrio Agung Wicaksono, & Nanang Yudi Setiawan (2019), dari Jurusan Sistem Informasi, Fakultas Ilmu Komputer, Universitas Brawijaya dengan judul Pengembangan Sistem Informasi Tracer Study Alumni Berbasis Web Menggunakan Metode RUP (Studi Kasus: SMA Suluh Jakarta Selatan). Hasil siswa SMA akan mempengaruhi cara mereka berpikir ketika mereka masuk ke perguruan tinggi, jadi penting untuk melakukan penilaian pembelajaran yang berkelanjutan.

Saat ini, SMA Suluh, salah satu sekolah menengah swasta di Jakarta, menghadapi masalah dalam mengelola data siswanya. Sebagai orang yang bertanggung jawab atas pendataan alumni, guru BK tidak dapat sepenuhnya mengetahui di mana alumni berada. Akibatnya, menghubungi alumni untuk mengikuti kegiatan di SMA Suluh tidak efektif karena mereka tidak memiliki informasi yang mereka butuhkan. Oleh karena itu, untuk membantu guru BK dengan masalah ini, sistem informasi alumni studi tindak lanjut harus dibuat. Dengan persentase hasil uji penerimaan pengguna tahap transisi sebesar 91,4%, sistem ini adalah sistem informasi berbasis web yang dikembangkan dengan metode RUP.

Penelitian ini menemukan bahwa manfaatnya termasuk lembaga pendidikan mengelola data alumni dengan lebih efisien dan alumni memiliki akses lebih mudah ke informasi dan jaringan alumni. Secara keseluruhan, penelitian ini mendukung peningkatan kualitas pendidikan dan hubungan antara lembaga pendidikan dan dunia kerja, Bagi pembaca literatur, informasi ini akan membantu mereka mengembangkan sistem informasi alumni dan kebijakan untuk mengelola data alumni.

2.1.10 Literatur 10

Oleh Aminurlah Syam, & Abdul Rachman Manga (2017) dari jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Muslim Indonesia dengan judul Sistem Tracer Study Alumni Fakultas Ilmu Komputer Universitas Muslim Indonesia Menggunakan Metode *On-Line Analytical Processing* (Olap). Dalam penelitian ini, penulis membuat sistem berbasis web yang menggunakan data warehouse dan menerapkan metode pengolahan analisis online (OLAP). Sistem ini kemudian akan berfungsi sebagai laporan data lulusan yang digambarkan dalam bentuk grafik.

Metode OLAP digunakan untuk menampilkan data warehouse dalam penelitian ini, yang menghasilkan laporan dalam bentuk grafik. Selain itu, sistem berbasis web memudahkan mahasiswa dan alumni untuk memasukkan data, karena dapat dilakukan di mana saja dengan koneksi internet.

Penelitian ini memberikan informasi bahwa dengan adanya sistem berbasis web, mahasiswa dapat melengkapi dan memperbarui data angket kapan saja mereka

inginkan. Admin yang dapat dengan mudah membuat laporan grafis yang menjelaskan kualitas alumni di dunia kerja, baik di industri maupun di luar industri.

2.2 Tracer Study

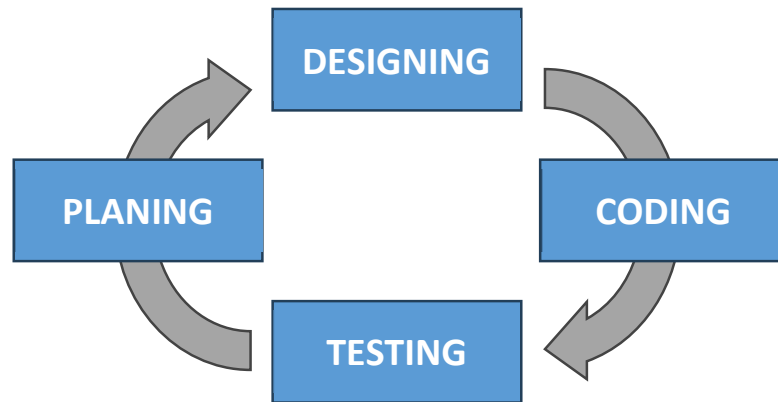
Tracer Study adalah studi penelusuran lulusan yang dilakukan beberapa tahun setelah kelulusan dan bertujuan untuk mengetahui hasil pendidikan seperti transisi dari perguruan tinggi ke dunia kerja, output keilmuan seperti penilaian diri tentang penguasaan dan pemerolehan kompetensi, proses pendidikan seperti evaluasi proses pembelajaran, kontribusi perguruan tinggi terhadap pemerolehan kompetensi, dan input pendidikan seperti penggalan informasi tentang pendidikan tinggi (Shelly Andari et al., 2021). Tujuan dari penelitian Tracer adalah untuk mengetahui capaian pendidikan seperti transisi dari pendidikan tinggi ke dunia kerja, capaian pendidikan seperti penilaian keterampilan, proses pendidikan seperti evaluasi proses pembelajaran, dan kontribusi pendidikan tinggi untuk perolehan keterampilan dan investasi. Sejak tahun 2011, Administrasi Umum Pendidikan Tinggi Kementerian Pendidikan telah menggunakan penelitian tracer untuk memantau bagaimana lulusan perguruan tinggi Indonesia beradaptasi untuk memulai karir mereka. Ini dilakukan selain untuk tujuan akreditasi (Aminullah Syam & Abdul Rachman Manga, 2017) .

2.3 Metode Pengembangan Sistem

2.3.1 Metode *Extreme Programming*

Menurut (Temi Ardiansah, 2022) Salah satu metode agile yang paling populer adalah *Extreme Programming*, terutama untuk proyek pengembangan aplikasi skala kecil. Ini disebabkan oleh fakta bahwa metode ini sangat mudah dan sederhana, tetapi tetap menerapkan berbagai prinsip agile, yang dianggap sebagai

kemajuan dalam meningkatkan efisiensi dan efektivitas pengembangan perangkat lunak. Metode *Extreme Programming* sangat cocok jika dihadapkan pada persyaratan yang tidak jelas dan perubahan yang terjadi dengan cepat.



Gambar 2.1 Proses Pengembangan Extreme Programming

Olah karena itu sebagai penyelesaian implementasi, metode extreme programming digunakan dengan tahapan berikut:

1. Tahapan *Planing*

Melakukan wawancara untuk mengetahui masalah apa yang terjadi, dan kemudian mengidentifikasi peluang, yaitu membantu pelanggan membuat rencana proyek yang ringkas dan fleksibel.

2. Tahapan *Design*

Menggunakan permodelan berorientasi objek untuk membuat rancangan sistem informasi, misalnya dengan membuat *Usecase Diagram*, *Class Diagram*, dan *Activity Diagram*.

3. Tahapan *Coding*

Tahapan coding adalah membuat aplikasi berbasis web dengan menggunakan bahasa pemrograman Javasript, NextJs sebagai framework, PostgreSQL sebagai databse, dan supabase sebagai *cloud storage*.

4. Tahapan *Testing*

Tahap di mana produk diuji, yang mencakup hasil pengujian sistem informasi. Pengujian ini dilakukan secara mandiri dengan partisipasi pengguna sampai hasilnya sempurna.

2.4 HTML

HTML, singkatan dari *HyperText Markup Language*, adalah bahasa markup umum yang digunakan untuk membuat struktur dan format halaman web. HTML terdiri dari berbagai tag yang digunakan untuk mendefinisikan berbagai elemen halaman web, seperti teks, judul, paragraf, gambar, dan tautan. Fungsinya adalah untuk membentuk struktur tampilan sebuah website dengan menggunakan HTML untuk menulis dan mengatur struktur konten. Selain itu, *World Wide Web Consortium* (W3C) menetapkan dan mengawasi HTML sebagai standar internet (Dicoding Intern, 2019).

2.5 Javascript

JavaScript, sebuah bahasa pemrograman yang populer dan fleksibel, memungkinkan pengembang membuat situs web, aplikasi web, dan banyak lagi. JavaScript, juga disebut sebagai bahasa pemrograman sisi klien, digunakan oleh browser web untuk membuat halaman web lebih menarik dan dinamis. Kemampuan serbagunanya membuatnya dapat digunakan untuk berbagai tugas seperti manipulasi DOM, animasi, dan validasi formulir. JavaScript masih menjadi salah satu bahasa pemrograman paling populer di dunia dan sangat mudah dipelajari (Bonjar Basumatary & Nishant Agnihotri, 2022).

2.6 Node.js

Node.js, dikembangkan oleh Ryan Dahl pada tahun 2009 dan disponsori oleh Joyent, perusahaan tempat ia bekerja, Node.js adalah platform perangkat lunak (PLATFORM AS A SERVICE, PaaS) yang menggunakan JavaScript sebagai bahasa pemrograman utama. Node.js memungkinkan pengembangan aplikasi web, jaringan, dan *server-side*. Menurut (Bonjar Basumatary & Nishant Agnihotri, 2022) Platform runtime *open-source* Node.js memungkinkan pengembang menjalankan kode JavaScript di luar browser web. Node.js dibangun di atas mesin JavaScript Chrome V8 dan menawarkan kecepatan dan efisiensi yang tinggi, serta orientasi pada peristiwa, yang membuatnya ideal untuk membangun aplikasi real-time dan skalabel. Dengan dukungan komunitas besar yang aktif, Node.js banyak digunakan untuk membangun berbagai jenis aplikasi, seperti server web, aplikasi real-time, dan alat baris perintah. Secara keseluruhan, Node.js adalah komponen penting dalam lingkungan pengembangan JavaScript.

2.7 NextJs

Next.js adalah *framework open-source* yang dibangun di atas React.js untuk membangun aplikasi web modern dengan performa tinggi dan SEO yang optimal. Next.js menawarkan berbagai fitur canggih yang dapat membantu developer membangun aplikasi web dengan lebih mudah dan efisien (nextjs.org, 2023). Menurut (Ballamudi et al., 2021) Next.js merupakan kerangka kerja aplikasi web *full stack*, yang telah membantu pengembang mengatasi kesulitan menggabungkan berbagai teknologi saat membuat aplikasi web. Dengan memungkinkan pengembangan *front-end* dan *back-end* dilakukan di satu tempat, Next.js mempercepat pengembangan produk dan membuat tugas pengembang lebih

mudah. Meskipun demikian, ada beberapa hal yang dapat ditingkatkan, terutama dalam hal efisiensi build produksi.

2.8 PostgreSQL

PostgreSQL adalah DBMS *open-source* paling canggih di dunia karena mendukung sebagian besar transaksi SQL, kontrol kompetitif, dan memiliki fitur kontemporer seperti kueri kompleks, *trigger*, *view*, dan *database transaction*. Selain itu, memungkinkan penambahan ekstensi tipe data, fungsi, operator, dan bahasa pemrograman (Viloria et al., 2019). PostgreSQL memiliki kemampuan untuk mengolah data dalam tabel yang berhubungan satu sama lain. Untuk berbagai aplikasi yang membutuhkan pengolahan data yang kompleks.

PostgreSQL memiliki fitur unggulan selain dukungan berbagai bahasa pemrograman. Misalnya, Asynchronous Replication, yang dapat menggandakan database secara asinkron, Mechanism Locking, yang mengamankan database, dan Savepoints, yang dapat mengatasi kesalahan pada transaksi yang kompleks. Tambahan lagi, PostgreSQL mendukung perintah non-relasional seperti JSON.

2.9 Cloud Storage

Cloud storage juga dikenal sebagai penyimpanan awan, adalah cara penyimpanan data digital dalam komputer server. Penyimpanan awan mencakup banyak server yang terletak di berbagai tempat, dan perusahaan hosting bertanggung jawab untuk memastikan bahwa data selalu dapat diakses dan bahwa lingkungan fisik tetap aman. Orang dan organisasi dapat menyewa atau membeli kapasitas penyimpanan yang berbeda dari pengelola penyimpanan cloud (Gunadi et al., 2020).

2.10 MVP (Minimum Viable Product)

Minimum Viable Product, atau MVP, adalah produk dengan beberapa fitur dasar yang dianggap cukup untuk menarik perhatian pengguna dan mendapatkan umpan balik dari target pasar. MVP tidak harus memiliki semua fitur yang diinginkan, tetapi cukup untuk menunjukkan nilai produk kepada pengguna (Geofanni Nerissa Arviana, 2023).

Manfaat dari pengembangan produk menggunakan konsep MVP:

- a. **Mengembangkan produk dengan cepat dan hemat biaya:** MVP memungkinkan Kita untuk fokus pada fitur-fitur yang paling penting dan menguji ide dengan cepat tanpa menghabiskan banyak waktu dan uang.
- b. **Mendapatkan feedback dari pengguna:** MVP membantu Kita untuk mendapatkan feedback dari pengguna awal dan menggunakan feedback tersebut untuk menyempurnakan produk Anda.
- c. **Mengurangi risiko kegagalan:** MVP membantu Kita untuk mengurangi risiko kegagalan produk dengan memvalidasi ide Kita sebelum Kita berinvestasi banyak waktu dan uang.

2.11 UML (Unified Modeling Language)

Menurut buku karya dari (Jacobson & Booch, 2021) bahasa pemodelan visual *Unified Modeling Language* (UML) adalah komponen penting dalam pengembangan sistem perangkat lunak karena membantu pemangku kepentingan proyek berkomunikasi lebih baik, meningkatkan pemahaman tentang sistem perangkat lunak, dan memberikan berbagai macam diagram untuk memodelkan struktur, perilaku, dan interaksi sistem tersebut. Dari berbagai macam diagram

permodelan dalam penelitian ini penulis hanya menggunakan Use Case Diagram, Activity Diagram, dan Class Diagram.

Table 2.2 Diagram Permodelan

Kategori	Diagram	Deskripsi
Struktur	Class Diagram	Menggambarkan struktur kelas dan hubungan antar kelas dalam sistem.
	Object Diagram	Menggambarkan snapshot objek dalam suatu sistem dan hubungannya.
	Component Diagram	Menggambarkan hubungan struktural antar elemen sistem sebuah software.
	Deployment Diagram	Menggambarkan arsitektur fisik sistem dan bagaimana komponen sistem di-deploy.
	Package Diagram	Menggambarkan hubungan antara paket-paket yang terdapat pada sebuah sistem.
Perilaku	Use Case Diagram	Menggambarkan interaksi antara pengguna dan sistem.
	Activity Diagram	Menggambarkan alur kerja atau alur aktivitas dalam sebuah sistem.
	Sequence Diagram	Menggambarkan interaksi antar objek dalam sistem dalam urutan waktu.
	State Machine Diagram	Menggambarkan perilaku sistem dalam berbagai keadaan.
	Timing Diagram	Menggambarkan interaksi antar objek dalam sistem dengan fokus pada waktu.
Interaksi	Collaboration Diagram	Menggambarkan interaksi antar objek dalam sistem.

Kategori	Diagram	Deskripsi
	Communication Diagram	Menggambarkan interaksi antar objek dalam sistem dengan fokus pada pesan yang dipertukarkan.

Secarakeseluruhan, UML adalah alat yang kuat untuk pengembangan sistem perangkat lunak yang membantu pengembang memahami, merancang, dan mendokumentasikan sistem dengan lebih baik. Semua jenis aplikasi perangkat lunak dapat dibangun dengan menggunakan UML, dan aplikasi ini dapat berjalan pada sistem operasi, jaringan, dan perangkat keras apa pun.

2.11.1 Use Case Diagram

Use Case Diagram menurut buku karya dari (Jacobson & Booch, 2021) Salah satu jenis diagram UML (*Unified Modelling Language*) yang digunakan untuk menggambarkan interaksi antara sistem dan pengguna adalah *Use Case Diagram*. Diagram ini membantu menunjukkan fungsionalitas sistem dan bagaimana pengguna akan berinteraksi dengannya.

Use Case Diagram dapat digunakan untuk mengidentifikasi dan menetapkan semua kemungkinan skenario penggunaan sistem, sehingga memastikan bahwa semua kebutuhan pengguna terpenuhi. Selain itu, *Use Case Diagram* juga membantu dalam merancang antarmuka pengguna yang intuitif dan mudah digunakan, dengan menggambarkan bagaimana setiap aksi atau fungsi akan dieksekusi oleh pengguna. Implementasi *Use Case Diagram* dalam pengembangan perangkat lunak dapat mempermudah tim pengembang dalam memahami tujuan bisnis dari sistem yang dibangun, sehingga meminimalkan kesalahan atau perubahan yang tidak diinginkan selama proses pembuatan aplikasi. Secara




keseluruhan, penerapan *Use Case Diagram* merupakan langkah penting dalam tahap analisis kebutuhan perangkat lunak agar proyek dapat berhasil dikembangkan sesuai dengan harapan dan kebutuhan para pengguna akhir.


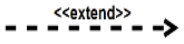
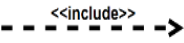
Beberapa elemen dari *Use Case Diagram* yang dikutip dari (Jacobson & Booch, 2021):

- a. Aktor: Entitas yang berinteraksi dengan sistem, bisa berupa manusia, perangkat lunak, atau sistem lain.
- b. *Use Case*: Sekelompok tindakan yang dilakukan oleh sistem untuk memberikan nilai kepada aktor.
- c. Hubungan: Asosiasi antara aktor dan *use case*, menunjukkan bagaimana aktor dapat menggunakan *use case*.

Berikut simbol yang digunakan dalam use case diagram dapat dilihat di tabel 2.2 dibawah ini.

Table 2.3 Use Case Diagram Symbol

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Mewakili peran manusia, sistem lain ataupun alat tertentu yang berkomunikasi dengan use case suatu system yang akan di bangun
2		<i>Use Case</i>	Abstraksi dan interaksi antara sistem dan actor biasanya dinyatakan dengan kata kerja
3		<i>Association</i>	Abstraksi dari penghubung antara <i>Actor</i> dengan <i>use case</i>

No	Simbol	Nama	Keterangan
4		<i>Generalisation</i>	Generalisasi adalah hubungan hierarkis antara jenis menggunakan di mana jenis menggunakan yang lebih umum (<i>superclass</i>) memiliki perilaku dan karakteristik yang sama atau hampir sama dengan jenis menggunakan yang lebih khusus (<i>subclass</i>).
5		<i>Extend</i>	<i>Extend</i> adalah hubungan antara dua kebutuhan, dengan kebutuhan yang lebih besar (<i>extended use case</i>) menunjukkan kebutuhan tambahan yang dapat dilakukan oleh kebutuhan yang lebih kecil.
6		<i>Include</i>	<i>Include</i> adalah hubungan antara dua <i>use case</i> , di mana <i>use case</i> yang lebih kecil (<i>included use case</i>) menggambarkan fungsionalitas yang terlibat atau terjadi dalam <i>use case</i> yang lebih besar (<i>including use case</i>).

2.11.2 Activity Diagram

Activity Diagram menurut buku karya dari (Jacobson & Booch, 2021) Salah satu jenis diagram UML yang digunakan untuk menggambarkan aliran aktivitas atau alur kerja dalam sebuah sistem adalah *Activity Diagram*. *Activity Diagram* membantu memvisualisasikan bagaimana berbagai aktivitas dalam sistem saling terkait dan bagaimana sistem akan berperilaku dalam situasi tertentu. Selain itu, diagram aktivitas membantu menemukan proses-proses yang berjalan secara bersamaan atau berurutan dalam sistem.

Activity Diagram dapat membantu tim pengembang perangkat lunak merancang dan mengimplementasikan sistem dengan lebih efisien dengan



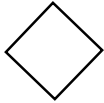
menggunakan simbol yang mudah dipahami. Selain itu, diagram ini dapat digunakan sebagai alat untuk berkomunikasi dengan pemangku kepentingan dan tim pengembang untuk memastikan bahwa semua orang memahami alur kerja sistem yang akan dibangun.



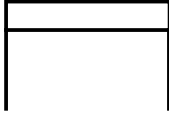
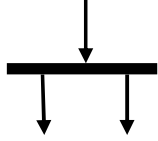
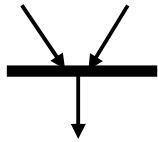
Elemen Activity Diagram:

- a. **Activity:** Sebuah langkah atau proses dalam sistem.
- b. **Action:** Sebuah tindakan yang dilakukan oleh suatu objek.
- c. **Control Flow:** Menunjukkan urutan dan kondisi di mana aktivitas dan action terjadi.
- d. **Object:** Entitas yang terlibat dalam aktivitas.

Sama seperti use case diagram, activity diagram juga memiliki symbol nya sendiri, seperti yang di tampilkan pada tabel 2.3 dibawah ini:

Table 2.4 Symbol Activity Diagram

No	Simbol	Nama	Keterangan
1		<i>Initial Node</i>	Ditunjukkan sebagai status awal sistem dan memulai proses pada diagram.
2		<i>Activity</i>	Mewakili langkah-langkah dalam proses. Semua aktivitas diberi nama dan digambarkan sebagai persegi panjang dengan sudut lengkung. Aktivitas awal adalah titik awal di mana proses berakhir.
3		<i>Action</i>	Jika ada lebih dari satu pilihan aktifitas, percabangan, atau keputusan, biasanya digunakan. Ini dapat menunjukkan situasi atau keputusan yang harus diambil selama proses.

No	Simbol	Nama	Keterangan
4		<i>Extend</i>	Symbol "penggabungan" biasanya digunakan untuk menyatukan dua atau lebih aktivitas menjadi satu.
5		<i>Activity Final Node</i>	Symbol Status Akhir digunakan ketika diagram aktifitas selesai. Ini menunjukkan bahwa proses telah berakhir.
6		<i>Swimlane</i>	Digunakan untuk memisahkan organisasi bisnis yang terlibat satu sama lain dan bertanggung jawab atas pekerjaan yang sedang berlangsung.
7		<i>Fork</i>	Aliran aktivitas dibagi menjadi beberapa jalur paralel dengan menggunakan <i>fork</i> .
8		<i>Join</i>	Sementara <i>join</i> digunakan untuk menggabungkan jalur paralel yang berbeda menjadi satu.

2.11.3 Class Diagram





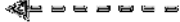
Class Diagram adalah salah satu jenis diagram UML yang paling penting dalam pemodelan sistem perangkat lunak adalah diagram kelas, yang menunjukkan struktur kelas dan hubungan antar kelas. Dalam diagram ini, kelas digambarkan sebagai blok yang menunjukkan jenis data atau objek dalam sistem, dengan atribut yang menunjukkan karakteristik atau properti kelas, dan metode yang menunjukkan fungsi atau operasi yang dapat dilakukan oleh kelas. Selain itu, hubungan antar



kelas dan kelas-kelas tersebut juga digambarkan sebagai blok yang menunjukkan jenis data atau objek (Jacobson & Booch, 2021).

Class Diagram memiliki beberapa element yaitu:

- a. **Kelas:** Sebuah blok yang mewakili jenis data atau objek dalam sistem.
- b. **Atribut:** Karakteristik atau properti dari sebuah kelas.
- c. **Metode:** Fungsi atau operasi yang dapat dilakukan oleh sebuah kelas.
- d. **Hubungan:** Asosiasi antara kelas, seperti *inheritance*, *association*, dan *aggregation*.

Table 2.5 Simbol-simbol Class Diagram

No	Simbol	Nama	Keterangan
1		<i>Association</i>	Yang menghubungkan satu objek dengan objek lainnya
2		<i>Nary Association</i>	Mengusahakan untuk tidak terlibat dengan lebih dari dua objek
3		<i>Class</i>	Kumpulan objek dengan atribut dan operasi yang sama
4		<i>Collaboration</i>	Urutan tindakan yang dilakukan oleh sistem yang memberikan hasil yang dapat diukur kepada aktor
5		<i>Realization</i>	Operasi yang sebenarnya dilakukan oleh objek

No	Simbol	Nama	Keterangan
6		<i>Dependency</i>	Hubungan di mana elemen yang bergantung pada elemen yang tidak independen akan dipengaruhi oleh perubahan pada elemen independen.
7		<i>Generalization</i>	Ketika objek anak (keturunan) berbagi perilaku dan struktur data dengan objek induk (<i>parent</i>)

2.12 Pengujian Black Box

Teknik pengujian dalam *black box*, juga dikenal sebagai pengujian perilaku, digunakan untuk mengamati input dan output suatu aplikasi tanpa mengetahui atau membedah struktur kode aplikasi. Proses pengujian ini biasanya dilakukan di akhir fase pengembangan perangkat lunak untuk mengetahui apakah aplikasi berjalan dengan baik dan juga untuk menemukan bug atau error yang dibuat selama proses pengembangan (Rony Setiawan, 2021). Metode *black box testing* memungkinkan pengujian dari sudut pandang pengguna akhir untuk memastikan bahwa aplikasi berfungsi dengan baik dan memenuhi kebutuhan pengguna.

Fungsionalitas aplikasi adalah fokus utama pengujian ini, tanpa mempelajari detail implementasi kode yang digunakan. Ini membantu menemukan kesalahan atau kesalahan yang mungkin terlewatkan selama proses pengembangan sehingga dapat diperbaiki sebelum aplikasi dirilis untuk pengguna akhir.