

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Dalam penyusunan tugas akhir ini, penulis terinspirasi dan mengacu pada beberapa penelitian sebelumnya yang berhubungan dengan latar belakang masalah dalam tugas akhir ini. Berikut adalah beberapa penelitian yang terkait dengan tugas akhir ini:

Penelitian pertama yang berjudul "Implementasi Teknik *Pomodoro* dan *Lockscreen* pada Aplikasi *Locktimer* Berbasis Android" bertujuan untuk mengatasi masalah menurunnya konsentrasi penglihatan mata pada mahasiswa selama belajar yang disebabkan oleh penggunaan komputer dalam waktu yang lama, yang dapat mengakibatkan *Computer Vision Syndrome (CVS)*. Solusi yang diusulkan dalam penelitian ini adalah dengan mengimplementasikan teknik *Pomodoro* untuk membagi durasi belajar dan istirahat dengan tepat, sehingga meningkatkan fokus dan produktivitas. Selain itu, aplikasi juga dilengkapi dengan fitur *lock screen* pada setiap *timer* untuk memberi jeda dari penggunaan komputer atau smartphone selama waktu istirahat, sehingga membantu mencegah terjadinya *CVS*. Penelitian ini menggunakan metode *Multimedia Development Life Cycle (MDLC)* dalam perancangan aplikasi. Hasil pengujian menunjukkan bahwa aplikasi beroperasi dengan baik di *Smartphone* [23].

Penelitian kedua yang berjudul "Perancangan Aplikasi Belajar Bersama menggunakan Teknik *Pomodoro* Berbasis *Web*" bertujuan untuk merancang aplikasi pembelajaran interaktif secara *real-time* yang memanfaatkan teknik *Pomodoro* sebagai metode *time management*. Masalah yang diatasi dalam penelitian ini adalah kurangnya fokus dan terganggunya konsentrasi pelajar akibat kebiasaan *multitasking* saat belajar. Solusi yang diusulkan adalah menggunakan teknik *Pomodoro* untuk melatih otak fokus dan meningkatkan produktivitas dalam pengerjaan tugas. Metode penelitian yang digunakan adalah *Multimedia Development Life Cycle (MDLC)* yang terdiri dari 6 tahap, yaitu *Concept, Design, Material Collecting, Assembly, Testing, dan Distribution*. Aplikasi belajar ini memungkinkan pelajar untuk berdiskusi secara langsung dengan peserta lainnya, dengan tujuan membantu pelajar menghindari distraksi dan meningkatkan interaksi dalam belajar. Hasil pengujian menunjukkan bahwa aplikasi berfungsi dengan baik di *browser* dan telah tersedia link untuk akses lebih mudah [24].

Penelitian ketiga yang berjudul "*Pomodoro Technique For Improving Students' Reading Ability During Covid-19 Pandemic*", bertujuan untuk menyelidiki bagaimana Teknik *Pomodoro* dapat meningkatkan kemampuan membaca siswa selama pembelajaran dari rumah selama pandemi Covid-19. Masalah yang diatasi dalam penelitian ini adalah perubahan pola dan proses

pengajaran pembelajaran akibat pandemi Covid-19 di Indonesia yang membuat durasi belajar menjadi faktor penting. Solusi yang diusulkan adalah menerapkan Teknik *Pomodoro* sebagai metode manajemen waktu untuk membantu meningkatkan kemampuan membaca siswa selama belajar dari rumah. Metode penelitian yang digunakan adalah desain quasi-eksperimental melibatkan enam puluh siswa dari Program Studi Manajemen dan Akuntansi dalam mata pelajaran Bahasa Inggris. Setengah dari siswa diajar menggunakan Teknik *Pomodoro* untuk manajemen waktu, sementara siswa lainnya diajar menggunakan durasi kontinu seperti jadwal yang telah ditetapkan. *Pre-test* dan *post-test* digunakan sebagai instrumen pengumpulan data, dan skor siswa dianalisis dengan uji *t-sample independent t-test* menggunakan MS Excel 2019. Hasil penelitian menunjukkan bahwa rata-rata skor siswa yang menggunakan Teknik *Pomodoro* meningkat secara signifikan. Dengan demikian, Teknik *Pomodoro* dapat direkomendasikan sebagai salah satu teknik manajemen waktu yang efisien dalam meningkatkan kemampuan membaca siswa selama pembelajaran dari rumah [25].

Pada penelitian keempat, Kota Madiun menerapkan serangkaian strategi pembelajaran yang efektif, antara lain Teknik *Pomodoro*, Teknik Catatan Cornell, dan Teknik Feynman. Implementasi kegiatan ini melibatkan integrasi pembelajaran *online* yang dilakukan menggunakan *zoom meeting* untuk mengajarkan beragam strategi pembelajaran, bersama dengan praktik *offline* di kelas untuk menerapkan dan memperkuat keterampilan tersebut. Tujuan kegiatan ini adalah untuk mengenalkan pengajar dan siswa mengenai pelaksanaan kegiatan belajar mengajar yang lebih tepat sasaran dan efisien. Temuan penelitian menunjukkan bahwa penerapan kombinasi teknik belajar *Pomodoro*, Cornell Notes, dan Feynman telah divalidasi secara empiris untuk meningkatkan kinerja tes, memfasilitasi pemahaman konsep yang rumit, meningkatkan konsentrasi selama belajar, meningkatkan retensi memori, dan meningkatkan kualitas pembelajaran secara keseluruhan [26].

Pada penelitian kelima, sulitnya menjaga fokus belajar bagi mahasiswa, terutama pada masa pandemi COVID-19 yang penuh dengan tantangan dan gangguan. Oleh karena itu, penelitian ini bertujuan untuk meningkatkan efektivitas belajar mahasiswa dengan menerapkan Teknik *Pomodoro*. Metode *Pomodoro* ini berfokus pada manajemen waktu, di mana mahasiswa mengatur waktu belajar dan istirahat dengan efisien untuk tetap menjaga kondisi jasmani dan rohani agar dapat fokus dalam belajar. Penelitian ini mencoba memanfaatkan fakta bahwa manusia rata-rata hanya dapat fokus dalam konsentrasi tinggi selama 25-30 menit, dan mengatur waktu belajar sesuai dengan *interval* tersebut untuk meningkatkan efektivitas belajar [27].

Pada penelitian keenam, adalah rendahnya pemahaman dan penguasaan mahasiswa Program Studi Pendidikan Bahasa dan Sastra Indonesia fokus pada tindak tutur representatif dan macam-macamnya. Untuk menjawab permasalahan

tersebut, maka dilakukan penelitian yang bertujuan untuk menjelaskan dan memahami makna, klasifikasi, dan penerapan tindak tutur representatif dalam video berjudul “Cara Belajar Menggunakan Teknik *Pomodoro*” di channel *YouTube* “*Raining Question Marks*”. ". Penelitian ini menggunakan teknik mendengarkan dan pendekatan deskriptif untuk menyelidiki 20 tindak tutur teladan dalam 8 kategori tindak tutur yang berbeda. Hasil penelitian menunjukkan bahwa mahasiswa dapat sepenuhnya memahami dan mengerti tentang definisi, jenis, dan penggunaan tindak tutur representatif melalui analisis yang dilakukan [28].

Pada penelitian ketujuh, masalah yang dikaji dalam penelitian ini adalah bahaya *attention residue* yang terjadi pada pembelajaran daring (*online*) selama pandemi Covid-19. *Attention residue* terjadi ketika terjadi perpindahan aktivitas kognitif dengan cepat, dimana peserta didik masih berfokus pada aktivitas sebelumnya meskipun telah beralih ke aktivitas berikutnya. Hal ini dapat mengancam fokus dan kualitas belajar peserta didik. Untuk mengatasi masalah ini, penelitian ini menggunakan Metode *Pomodoro* yang berbasis pengaturan waktu belajar dan istirahat sebagai alternatif untuk meningkatkan hasil belajar peserta didik. Hasil penelitian menunjukkan bahwa setelah diterapkan Metode *Pomodoro*, peserta didik menunjukkan peningkatan hasil belajar yang lebih baik berdasarkan hasil *post-test* yang lebih tinggi dibandingkan dengan hasil *pre-test* [29].

Tabel 2. 1 Pemetaan Hasil Penelitian Terdahulu

Nama	Judul	Tahun	Metode Penelitian	Hasil Penelitian	Perbedaan
Dihin Muriyatmoko, Triana Harmini, Abdul Rohman [23]	Implementasi Teknik <i>Pomodoro</i> dan <i>Lockscreen</i> pada Aplikasi <i>Locktimer</i> Berbasis Android	2022	MDLC	Aplikasi beroperasi dengan baik di <i>Smartphone</i> , membantu membagi durasi belajar dan istirahat dengan tepat serta mencegah terjadinya <i>Computer Vision Syndrome (CVS)</i> .	Penelitian saya bertujuan mengatasi keterbatasan situs <i>web Pomodoro</i> seperti kurangnya kustomisasi warna dan rekomendasi <i>timer</i> , serta tidak ada fitur daftar tugas. Solusinya adalah mengembangkan situs <i>Pomodoro</i> dengan arsitektur <i>serverless</i> agar pengguna dapat kustomisasi <i>platform</i> sesuai preferensi mereka. Penelitian ini bertujuan mengatasi masalah konsentrasi mata pada mahasiswa selama belajar dengan menggunakan teknik <i>Pomodoro</i> dan <i>lock screen</i> pada aplikasi berbasis Android.
Fernando, Vieri [5]	Perancangan Aplikasi Belajar Bersama menggunakan Teknik <i>Pomodoro</i> Berbasis <i>Web</i>	2022	<i>Multimedia Development Life Cycle (MDLC)</i>	Aplikasi berfungsi dengan baik di browser, memungkinkan pelajar untuk berdiskusi secara langsung dengan peserta lainnya, dan telah menggunakan teknik <i>Pomodoro</i> untuk membantu meningkatkan fokus dan produktivitas pelajar.	Perbedaan penelitian saya adalah mengatasi keterbatasan pengguna situs <i>Pomodoro</i> terkait kustomisasi warna dan <i>timer</i> , serta kekurangan fitur daftar tugas. Solusinya adalah mengembangkan situs dengan arsitektur <i>serverless</i> . Sementara penelitian ini merancang aplikasi belajar interaktif dengan fokus mengatasi masalah kurangnya fokus dan gangguan saat belajar, menggunakan teknik <i>Pomodoro</i> sebagai metode manajemen waktu.

Nama	Judul	Tahun	Metode Penelitian	Hasil Penelitian	Perbedaan
Kisno Kisno [4]	<i>Pomodoro Technique For Improving Students' Reading Ability During Covid-19 Pandemic</i>	2020	<i>Desain quasi-eksperimental</i>	Penelitian ini memberikan rekomendasi agar Teknik <i>Pomodoro</i> digunakan sebagai salah satu teknik manajemen waktu pembelajaran yang efisien dalam meningkatkan kemampuan membaca mahasiswa selama pembelajaran dari rumah selama pandemi COVID-19.	Perbedaan dari penelitian saya adalah fokus pada pengembangan situs <i>Pomodoro</i> dengan arsitektur <i>serverless</i> untuk mengatasi keterbatasan kustomisasi dan fitur <i>todo list</i> . Sedangkan penelitian ini berfokus pada penerapan Teknik <i>Pomodoro</i> sebagai metode manajemen waktu untuk meningkatkan kemampuan membaca siswa selama pembelajaran dari rumah selama pandemi Covid-19.
Heidy Arviani, Dyva Claretta, Zainal Abidin Achmad [26]	Peningkatan Kualitas Belajar Siswa dengan Teknik <i>Pomodoro</i> , Cornell Notes dan Feynman di Sanggar Belajar Professor Kota Madiun	2021	Pengabdian Masyarakat dengan metode komprehensif yang menggabungkan Teknik <i>Pomodoro</i> , Teknik Cornell Notes, dan Teknik Feynman dengan tutorial daring dan praktik luring.	Penerapan teknik belajar <i>Pomodoro</i> , Cornell Notes, dan Feynman dapat meningkatkan nilai ujicoba ujian, memudahkan pemahaman konsep, memperkuat fokus belajar, meningkatkan daya ingat, dan meningkatkan kualitas belajar siswa.	Perbedaan dari penelitian saya adalah fokus pada pengembangan situs <i>Pomodoro</i> dengan arsitektur <i>serverless</i> untuk mengatasi keterbatasan kustomisasi dan fitur <i>todo list</i> . Sedangkan penelitian ini berfokus pada penerapan berbagai metode belajar, termasuk Teknik <i>Pomodoro</i> , Teknik Cornell Notes, dan Teknik Feynman, untuk meningkatkan kualitas belajar siswa di Sanggar Belajar "Professor" Kota Madiun. Hasil penelitian menunjukkan bahwa penerapan teknik-teknik belajar tersebut berhasil meningkatkan nilai ujian, memudahkan pemahaman konsep, memperkuat fokus belajar, meningkatkan daya ingat, dan meningkatkan kualitas belajar siswa.

Nama	Judul	Tahun	Metode Penelitian	Hasil Penelitian	Perbedaan
Muhammad Zaki Zikrillah Baharuddin Nasution, Muhammad Irwan Padli Nasution, Sri Suci Ayu Sundari [27]	PENERAPAN TEKNIK <i>POMODORO</i> DALAM UPAYA MENINGKATKAN EFEKTIVITAS BELAJAR MAHASISWA PADA MASA PANDEMI COVID-19 DI KELAS SISTEM INFORMASI-3	2022	Sosialisasi dengan penerapan teknik <i>Pomodoro</i> pada kelas Sistem Informasi-3 melalui media <i>ZOOM</i> dan <i>platform YouTube</i> .	Penerapan teknik <i>Pomodoro</i> dalam belajar efektif dapat meningkatkan kualitas belajar mahasiswa, termasuk daya ingat, konsentrasi, dan pemahaman ilmu pengetahuan. Mahasiswa yang menggunakan teknik <i>Pomodoro</i> mengalami peningkatan minat dan semangat dalam pelajaran serta dapat fokus dalam belajar.	Perbedaan dari penelitian saya adalah fokus pada pengembangan <i>website Pomodoro</i> dengan <i>serverless architecture</i> untuk mengatasi keterbatasan kustomisasi dan fitur <i>todo list</i> . Sementara itu, penelitian ini berfokus pada penerapan Teknik <i>Pomodoro</i> untuk meningkatkan efektivitas belajar mahasiswa dengan mengatur waktu belajar dan istirahat secara efisien, memanfaatkan fakta bahwa manusia cenderung fokus selama 25-30 menit untuk meningkatkan efektivitas belajar.
Frissillia Anzalina Devy, Asep Purwo Yudi Utomo [28]	Analisis Tindakan Tutur Representatif dalam Video "Cara Belajar dengan Teknik <i>Pomodoro</i> " Padakanal Youtube Hujan Tanda Tanya	2021	Metode Simak dan Metode Deskriptif	Mahasiswa Prodi Pendidikan Bahasa dan Sastra Indonesia dapat sepenuhnya memahami serta mengerti tentang definisi, jenis, dan penggunaan tindakan tutur representatif melalui analisis yang dilakukan pada video "Cara Belajar dengan Teknik <i>Pomodoro</i> " pada kanal <i>YouTube</i> "Hujan Tanda Tanya".	Perbedaan dari penelitian saya adalah fokus pada pengembangan <i>website Pomodoro</i> dengan <i>serverless architecture</i> dan fitur kustomisasi, sedangkan penelitian ini berfokus pada pemahaman dan penggunaan tindakan tutur representatif oleh mahasiswa Prodi Pendidikan Bahasa dan Sastra Indonesia melalui analisis pada video "Cara Belajar dengan Teknik <i>Pomodoro</i> " di kanal <i>YouTube</i> "Hujan Tanda Tanya".

Nama	Judul	Tahun	Metode Penelitian	Hasil Penelitian	Perbedaan
Tarwiyah Tarwiyah, R. John Pieter Simarmata, Conrad Bombongan [29]	Metode <i>Pomodoro</i> untuk Mengatasi <i>Attention Residue</i> pada Masa Pembelajaran Daring	2021	Metode <i>Pomodoro</i>	Hasil penelitian menunjukkan bahwa setelah diterapkan Metode <i>Pomodoro</i> , peserta didik menunjukkan peningkatan hasil belajar yang lebih baik berdasarkan hasil <i>post-test</i> yang lebih tinggi dibandingkan dengan hasil <i>pre-test</i> .	Perbedaan dari penelitian saya adalah fokus pada pengembangan <i>website Pomodoro</i> dengan <i>serverless architecture</i> dan fitur kustomisasi, sementara penelitian ini berfokus pada bahaya <i>attention residue</i> yang muncul selama pembelajaran daring (<i>online</i>) selama pandemi Covid-19. Penelitian kedua menggunakan Metode <i>Pomodoro</i> sebagai alternatif untuk meningkatkan hasil belajar peserta didik dengan berbasis pengaturan waktu belajar dan istirahat. Hasil penelitian menunjukkan peningkatan hasil belajar yang lebih baik setelah penerapan Metode <i>Pomodoro</i> .

Adapun perbedaan dari penelitian diatas adalah fokus dan tujuan penelitian. Penelitian ini bertujuan untuk mengatasi masalah terkait dengan keterbatasan penggunaan situs *web Pomodoro*, seperti kurangnya opsi kustomisasi untuk warna dan rekomendasi *timer*. Penelitian ini juga ingin menghadirkan fitur daftar tugas (*todo list*) yang seringkali tidak tersedia di situs *web Pomodoro* lainnya. Solusi yang diusulkan adalah mengembangkan situs *web Pomodoro* dengan menggunakan arsitektur *serverless*, yang memberikan kebebasan bagi pengguna untuk menyesuaikan *platform* sesuai dengan preferensi mereka. Peneliti ingin menciptakan lingkungan belajar yang serba guna yang memenuhi preferensi individu, meningkatkan pengalaman belajar untuk setiap pengguna berdasarkan kebutuhan unik mereka. Selain itu, integrasi fitur kustomisasi, seperti warna, rekomendasi *timer*, dan daftar tugas (*todo list*), akan memberdayakan pengguna untuk menciptakan pengalaman *Pomodoro* yang *personal* yang mengoptimalkan efisiensi belajar dan membantu mempertahankan fokus selama sesi belajar.

2.2. Landasan Teori

2.2.1 Metode *Pomodoro*

Teknik *Pomodoro* adalah sebuah teknik pengaturan waktu kerja yang diciptakan oleh Francesco Corillo. Nama *Pomodoro* diambilnya dari *timer* dapur yang berbentuk tomat. Dalam Bahasa Italia, *Pomodoro* berarti tomat. Teknik ini sangat terkenal untuk meningkatkan produktivitas. Prinsipnya sederhana saja, yaitu memecah waktu kerja menjadi beberapa sesi. Anda bebas menentukan lamanya waktu bekerja dalam satu sesi. Idealnya, setiap sesi terdiri dari 25 menit. Setelah 25 menit tersebut usai, Anda berhak atas waktu istirahat sebanyak 5 menit. Setelah empat kali sesi selesai (4 X 25 menit), waktu istirahat yang bisa Anda peroleh naik, menjadi 15 menit [30].

Teknik *Pomodoro* merupakan metode manajemen waktu yang dikembangkan oleh Francesco Cirillo dari Italia. Pada awal kuliahnya pada akhir tahun 1980, Francesco mengalami kesulitan dalam menjaga konsentrasi saat belajar. Namun, ia akhirnya menyadari bahwa ia perlu menerapkan rutinitas belajar yang mencakup interval waktu tertentu untuk belajar, diselingi dengan momen relaksasi, untuk memulihkan kesehatan fisik dan psikologisnya. Kata "*Pomodoro*" adalah istilah Italia untuk tomat. Nama tersebut dipilih karena Francesco menggunakan pengatur waktu dapurnya, yang kebetulan bentuknya menyerupai tomat merah, selama pengembangan pendekatan ini [31].

Kemudian Teknik *Pomodoro* merupakan sebuah metode belajar yang lebih fokus terhadap manajemen waktu, di mana pelajar memiliki waktu 25 menit di dalam proses belajar yang terus dilakukan berulang kali. Untuk pembagiannya adalah, 20 menit digunakan untuk belajar dan 5 menit digunakan untuk beristirahat sejenak. Setelah 25 menit, pelajar mengulangnya lagi. Setelah 4 kali belajar, pelajar harus beristirahat selama 30 menit. Metode belajar ini dapat disesuaikan dengan kemampuan berapa lama pelajar dapat bertahan untuk belajar. Jika pelajar hanya dapat belajar selama 15 menit, waktu yang 10 menitnya digunakan untuk beristirahat [32].

2.2.2 Website

Website adalah kumpulan dokumen berupa halaman *web* yang memuat teks dalam format *Hyper Text Markup Language (HTML)*. Sebuah *website* disimpan pada *server hosting* yang dapat diakses menggunakan browser dengan koneksi internet menggunakan alamat internet yang disebut *Uniform Resource Locator (URL)* [33]. *Website* terdiri dari dua jenis yaitu *website* statis dan *website* dinamis. *Website* statis adalah *website* yang jika ingin diubah konten didalamnya harus mengeditnya secara manual, artinya harus merubah *source codenya*. Biasanya

halaman *website* statis masih menggunakan *tag HTML* dan datanya belum tersimpan di database. Sedangkan *website* dinamis, jika ingin mengubah konten pada *website* dapat dilakukan dengan mudah tanpa harus membuka *source code* dan dapat diperbarui secara berkala. Hal ini karena konten *website* disimpan dalam *database*. Halaman *website* dinamis biasanya ditulis dalam bahasa pemrograman *server side* seperti *PHP, ASP, JSP* dan lain-lain [33].

Namun, situs *web* juga dapat didefinisikan sebagai kumpulan halaman *web* dalam *domain* tertentu yang berisi informasi. Sebuah situs *web* biasanya terdiri dari beberapa halaman *web* yang saling berhubungan. *Hyperlink* mengacu pada hubungan antara dua halaman *web*, sedangkan teks yang berfungsi sebagai media penghubung dikenal sebagai *hypertext* [34].

2.2.3 *Serverless Computing* / *Arsitektur Serverless*

Serverless Computing adalah model di mana penyedia *cloud* secara otomatis mengelola infrastruktur yang dibutuhkan untuk menjalankan kode [35], sehingga pengguna tidak perlu khawatir tentang pengaturan dan pemeliharaan *server*. Ini memungkinkan pengembangan untuk *focus* pada penulisan kode dan mempercepat waktu pengembangan. Contoh dari teknologi ini adalah *AWS Lambda, Google Cloud Functions, Vercel, dan Azure Functions*.

Function as a service (FaaS), juga dikenal sebagai “*serverless computing*”, adalah jenis *cloud computing* yang *focus* pada menjalankan kode atau fungsi tertentu sebagai *respons* terhadap suatu peristiwa atau permintaan. Dalam model ini, pengguna hanya perlu menyediakan kode fungsi, sementara penyedia *cloud* mengelola infrastruktur dan skala otomatis untuk menjalankan fungsi tersebut. Contoh penyedia *FaaS* termasuk *AWS Lambda, Azure Functions, Vercel, dan Google Cloud Functions* [36].

2.2.4 *Javascript*

JavaScript adalah bahasa pemrograman yang awalnya dirancang untuk berjalan di atas *browser web*. Namun, seiring berjalannya waktu, *JavaScript* tidak lagi terbatas hanya berjalan di *browser* saja. *Javascript* juga dapat digunakan di *Server, Game, IoT, Desktop, dan platform* lainnya. Awalnya, bahasa pemrograman yang sekarang dikenal sebagai *Javascript* diberi nama *Mocha*. nama ini kemudian diubah menjadi *LiveScript* ketika versi *beta browser Netscape Navigator 2.0* dirilis pada bulan September 1995. Namun kemudian diubah namanya menjadi *Javascript*. *Microsoft* mengadopsi teknologi serupa, terinspirasi oleh kesuksesan *Javascript*. *Microsoft* telah mengembangkan versi '*Javascript*' sendiri yang disebut *Script*. Kemudian diinstal di *Internet Explorer 3.0*. Hal ini menyebabkan terjadinya

'perang browser', karena *Script* yang dimiliki oleh *Microsoft* berbeda dengan *Javascript* yang dikembangkan oleh *Netscape*. Pada tahun 1996, *Netscape* akhirnya menyerahkan standarisasi *ECMA-262* ke *Ecma International*. Maka lahirlah standarisasi kode *JavaScript* yang diberi nama *ECMAScript* atau *ES*. Saat ini *ECMAScript* telah mencapai versi 10 (*ES10*) [37].

JavaScript adalah bahasa skrip yang banyak digunakan di internet dan kompatibel dengan browser web umum seperti *Internet Explorer (IE)*, *Mozilla Firefox*, *Netscape*, dan *Opera*. *Elemen SCRIPT* digunakan untuk memasukkan kode *JavaScript* ke halaman web [38]. *Javascript* juga Bahasa skrip yang berbasis objek. Artinya saat penulisan kode *JavaScript*, Anda menggunakan objek-objek bawaan atau objek-objek yang Anda ciptakan sendiri [39].

2.2.5 Next.js

Next.js adalah kerangka kerja *open-source* yang populer untuk membangun aplikasi *React* dengan *server-side rendered (SSR)* dan *static site generated (SSG)*. Dikembangkan oleh *Vercel*, *Next.js* menyediakan pendekatan yang intuitif dan ramah pengembang untuk membangun aplikasi web, memungkinkan pengembang untuk fokus pada pembuatan pengalaman pengguna yang hebat tanpa khawatir tentang infrastruktur yang mendasarinya [40].

Dengan *Next.js*, pengembang dapat membangun aplikasi web yang dinamis dan berkinerja tinggi yang dapat dijalankan di server atau platform mana pun, berkat kemampuan *rendering* universalnya [40]. *Next.js* menawarkan berbagai fitur dan manfaat, termasuk:

- *Server-side rendering (SSR)* dan *static site generation (SSG)*
- *Built-in tooling* dan optimisasi otomatis
- *Automatic code splitting* dan optimisasi
- Pendekatan *hybrid*
- *Setup* dan *deployment* yang mudah
- Dukungan komunitas dan ekosistem

2.2.6 Vercel

Seperti *Netlify*, *Vercel* (sebelumnya dikenal sebagai *Zeit*) menawarkan platform cloud untuk *static sites* seperti situs *Gatsby* dan mekanisme berbasis pemicu dari penyedia *Git* untuk melakukan *continuous deployment*. Penawaran *Vercel* mencakup *autoscaling*, *global edge network*, *asset compression*, *cache invalidation*, dan pengalaman pengembang tanpa konfigurasi [41].

Seperti *Netlify*, *Vercel* paling kuat ketika diintegrasikan langsung dengan repositori *Git* menggunakan *native Import Flow Vercel* setelah menghubungkan integrasi ke *GitHub*, *GitLab*, atau *Bitbucket*. Setiap push kode akan memicu *build* dan *deployment* baru yang dapat ditentukan secara arbitrer untuk terjadi hanya pada *branch* tertentu atau semua *branch*. Ini akan menghasilkan apa yang disebut *Vercel* sebagai *preview deployments*, dan semua modifikasi yang dilakukan pada *branch master* atau *main* akan memfasilitasi *production deployments*. *Vercel* juga menawarkan dukungan *domain* kustom melalui halaman *Domain Settings* pemilik akun [41].

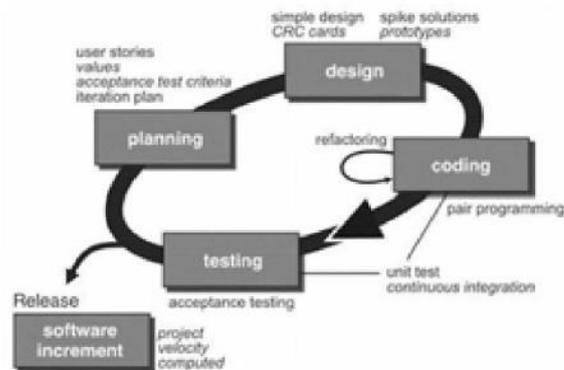
2.2.7 Tailwind

Menurut buku “*Tailwind CSS*” oleh Ivaylo Gerchec (2022), *Tailwind* adalah kumpulan kelas utilitas tingkat rendah yang dapat digunakan sebagai blok pembangunan untuk menciptakan hampir semua desain yang kita bayangkan. Kerangka kerja utilitas ini mencakup properti *CSS* paling penting, namun mudah diperluas dengan berbagai cara. *Tailwind* dapat digunakan baik untuk prototipe cepat maupun untuk membuat desain yang lengkap. *Tailwind* memiliki dokumentasi yang sangat baik, mencakup setiap kelas utilitas secara detail dan menunjukkan cara-cara untuk disesuaikan [42].

2.2.8 Extreme Programming

XP (*eXtreme Programming*) kependekan dari *eXtreme Programming*, adalah pendekatan rekayasa perangkat lunak yang lazim digunakan oleh pengembang untuk pengembangan aplikasi. Kent Beck menemukan konsep *Extreme Programming* (*XP*) ketika dia ditugaskan untuk mengelola proyek penggajian bernama *C3* (*Chrysler Comprehensive Compensation*) di Chrysler. Inisiatif ini dimulai pada bulan Maret 1996. Proyek ini menghadapi risiko kegagalan karena rumitnya sistem yang sedang dikembangkan dan menemui kesulitan selama tahap pengujian sistem. Chrysler baru saja menunjuk Kent Beck sebagai konsultan rekayasa perangkat lunak. Selanjutnya, ia mendapat pengakuan sebagai nenek moyang *XP* [43].

Kent Beck bersama rekannya, yaitu Ron Jeffries, diberi kewenangan oleh Chrysler untuk melakukan berbagai perubahan di proyek *C3* untuk membuat proyek tersebut lebih efisien, lebih fleksibel. Akhirnya Kent Beck dan Ron Jeffries berhasil menvelesaikan proyek *C3* dengan menerapkan berbagai metode pada saat pengembangannya.



Gambar 2. 1 Metodologi *eXtreme Programming (XP)*

Banyak pengembang perangkat lunak sering menggunakan metodologi *Pemrograman eXtreme* untuk mengembangkan perangkat lunak dengan cepat. Tahapan pengembangan perangkat lunak menggunakan *XP* antara lain: Perencanaan, Desain, *Coding*, dan Pengujian.

Serangkaian teknik yang digunakan untuk mengelola proyek C3 diberi nama metodologi *eXtreme Programming (XP)*. *XP* kurang inovasi dan gagal memberikan kontribusi signifikan terhadap pengembangan aplikasi. Kent Beck mengakui dan menggarisbawahi bahwa *XP* tidak berlaku secara universal untuk semua proyek pengembangan perangkat lunak. Namun demikian, *XP* memiliki keuntungan karena cocok untuk proyek dengan kebutuhan dinamis atau proyek dengan beragam persyaratan yang tidak ditentukan secara eksplisit oleh pelanggan. *XP* sangat ideal untuk pengembangan proyek yang memerlukan adaptasi cepat terhadap perubahan yang muncul selama pengembangan aplikasi. *XP* sangat cocok untuk tim kecil dan dekat secara geografis yang terlibat dalam pengembangan system [43]. *Extreme Programming* memungkinkan untuk menjaga biaya perubahan perangkat lunak agar tidak naik secara drastis seiring berjalannya waktu [44].

2.2.9 *VSCode*

VSCode adalah *text editor open source* yang populer dan serbaguna. *VSCode* memiliki banyak fitur yang berguna bagi pengembangan perangkat lunak, *debugging*, *pengeditan kode*, dan integrasi *GIT* [45]. *VS Code* mendukung berbagai bahasa pemrograman dan fitur, termasuk *debugging*, *syntax highlighting*, *code completion*, dan integrasi *Git*. Salah satu fitur luar biasa dari *Visual Studio Code* adalah pasarannya yang luas, yang menawarkan berbagai ekstensi dan *plugin* yang dapat meningkatkan fungsionalitas *editor*. Ekstensi ini dapat digunakan untuk menambahkan dukungan untuk bahasa pemrograman tambahan, menyediakan alat

debugging tambahan, dan banyak lagi. *Visual Studio Code* juga memiliki terminal bawaan, yang memungkinkan pengembang dengan menjalankan perintah dan berinteraksi dengan kode mereka tanpa meninggalkan editor. Ini bisa sangat berguna untuk tugas seperti membangun dan menerapkan aplikasi [46]. Biasanya, perangkat lunak *Microsoft* memerlukan pembayaran. Meskipun demikian, program *VSCode* ini gratis, artinya bersifat *open source*. *VSCode* juga memiliki fungsionalitas *Split Editor* yang mirip dengan alat *Multi Select* di *Sublime Text*. Hal ini memungkinkan Anda mengakses dan memeriksa banyak *file* secara bersamaan, menghilangkan kebutuhan untuk terus-menerus berpindah antar *file* [47].

Ikuti Langkah-langkah berikut untuk menginstal VS. Code:

- Buka web <https://code.visualstudio.com/> dan download sesuai dengan sistem operasi komputer anda.
- Setelah berhasil didownload, *Double* klik (klik 2x) hasil *download* tadi untuk menginstall *VSCode* nya.
- Pada bagian *License Agreement* klik *I accept the agreement* > klik *Next*.
- Pada bagian *Select Destination Location* biarkan secara *default* > klik *Next*.
- Pada bagian *Select Start Menu Folder* > klik *Next*.
- Pada bagian *Select Additional Tasks* > klik *Next*.
- Pada bagian *Ready to Install* > klik *Install*.
- Tunggu proses *Installing* selesai.
- Setelah itu klik *Finish*.

2.2.10 Figma

Figma adalah aplikasi desain grafis khusus untuk mendesain *prototype*, *user interface*, dan antar muka aplikasi. Mengapa kita membutuhkan *Figma* ? Pada saat Anda ingin membuat aplikasi, baik berbentuk aplikasi *smartphone* ataupun sebuah *website*, maka langkah pertama yang seharusnya dilakukan adalah membuat desain *prototype* terlebih dulu [48]. *Figma* adalah salah satu *tools* atau aplikasi desain yang dapat digunakan pada *windows* dan *Mac OS* untuk membuat *prototype* aplikasi serta berbagai desain lainnya. Aplikasi ini berbasis *vektor*, sehingga memang sangat cocok digunakan untuk membuat *user interface* aplikasi atau *website* [49].

2.2.11 Use Case Diagram

Use Case Diagram adalah representasi grafis yang menggambarkan interaksi antara aktor (pengguna atau sistem lain) dengan sistem yang sedang dikembangkan, menunjukkan fungsi atau layanan yang disediakan dari sudut pandang pengguna. Diagram ini mencakup elemen-elemen utama seperti aktor, *use case*, hubungan, dan batas sistem, membantu menggambarkan bagaimana pengguna berinteraksi dengan sistem untuk mencapai tujuan tertentu. Dengan menampilkan skenario penggunaan dan batasan sistem, *Use Case Diagram* memfasilitasi pemahaman kebutuhan fungsional sistem dan mendukung pengembang serta pemangku kepentingan dalam mengidentifikasi dan mendefinisikan persyaratan sistem pada tahap awal pengembangan perangkat lunak.

Use Case menceritakan bagaimana sistem dan aktor-aktornya berkolaborasi untuk mencapai tujuan tertentu. Ini adalah deskripsi langkah demi langkah tentang cara tertentu dalam menggunakan sistem. Struktur dari sebuah *use case* pada dasarnya bersifat naratif. Cerita yang diceritakan adalah tentang bagaimana sistem dan aktor-aktornya bekerja sama untuk mencapai sesuatu yang berarti bagi aktor yang terlibat. Kolaborasi ini berbentuk dialog antara sistem dan aktor-aktornya, dengan semua pihak berkontribusi pada penyelesaian *use case* tersebut [50]. Adapun elemen-elemen pada *Use Case* yang dijelaskan pada tabel 2.2.

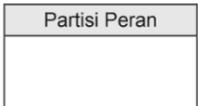
Tabel 2. 2 Elemen Use Case

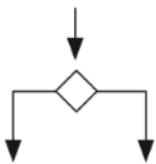
Elemen	Penjelasan
 An Actor	<i>Actor</i> mendefinisikan peran yang dapat dimainkan oleh pengguna saat berinteraksi dengan sistem. Pengguna bisa berupa individu atau sistem lain.
 A Use Case	<i>Use Case</i> menggambarkan interaksi antara aktor dan sistem, menguraikan bagaimana aktor memanfaatkan sistem untuk mencapai tujuan tertentu, dan merinci peran sistem dalam memfasilitasi pencapaian tujuan tersebut. Ini menggambarkan kolaborasi antara sistem dan aktor-aktornya untuk memberikan sesuatu yang berharga kepada setidaknya salah satu aktor.
	Elemen ini berfungsi untuk menghubungkan antara <i>actor</i> dan <i>use case</i>
	Elemen ini opsional, <i>tetapi</i> jika digunakan, panah menunjukkan elemen mana yang memulai interaksi. Pihak yang memulai berada di ujung tumpul garis.

2.2.12 Activity Diagram

Activity Diagram pada dasarnya digunakan untuk menggambarkan alur kerja (*workflow*) dari proses bisnis. Namun pada aplikasinya kita dapat menggunakannya untuk menjelaskan navigasi *user interface* atau antar muka dari aplikasi yang akan dibuat [51]. Diagram aktivitas memfasilitasi visualisasi dan pemahaman proses yang rumit dengan menggunakan simbol-simbol seperti aktivitas, garis penghubung, keputusan, percabangan, dan gabungan. Diagram aktivitas adalah jenis diagram tertentu di dalam *Unified Modeling Language (UML)* yang berfungsi untuk menggambarkan urutan tindakan atau tugas yang terlibat dalam proses bisnis, algoritma, atau fungsi sistem. Grafik ini mengilustrasikan urutan tindakan yang terjadi dalam suatu proses, bersama dengan urutan dan kondisi spesifik di mana aktivitas tersebut dijalankan.

Tabel 2. 3 Elemen Activity Diagram

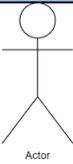
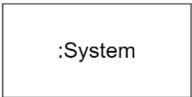
Elemen	Penjelasan
	Menunjukkan titik awal dari urutan tindakan atau aktivitas yang ada
	Elemen ini digunakan untuk menyatakan suatu tindakan dan bisa juga menetapkan tahapan aksi dari pelaku
	Mengilustrasikan perkembangan dari awal hingga akhir, seperti diagram alur yang komprehensif.
	Menunjukkan titik akhir dari urutan tindakan atau aktivitas yang ada
	Menunjukkan tingkat atau jangkauan suatu posisi. Pendekatan ini memungkinkan adanya klarifikasi alokasi tanggung jawab untuk setiap aktivitas individu.

Elemen	Penjelasan
	Digunakan untuk memfasilitasi percabangan pada diagram alir

2.2.11 Sequence Diagram

Diagram urutan sistem (*SSD*) adalah representasi grafis yang menggambarkan *input* dan *output*, serta interaksi berurutan, antara pengguna dan sistem dalam kasus penggunaan. Sequence diagram terdiri dari beberapa elemen yang dijelaskan pada tabel 2.4 [52]

Tabel 2. 4 Tabel Sequence Diagram

Elemen	Penjelasan
	<i>Actor</i> mewakili seorang aktor (orang atau peran yang berinteraksi dengan sistem)
	<i>System</i> adalah objek yang mewakili keseluruhan sistem yang terotomatisasi
	Garis hidup dalam Diagram Urutan mewakili objek, apakah itu aktor atau item, dan digambarkan sebagai garis putus-putus vertikal yang memanjang sepanjang rentang diagram.
	Anak panah antara <i>lifeline</i> mewakili <i>message</i> yang dikirim atau diterima oleh aktor dari sistem
	<i>Message</i> dikategorikan menggunakan label untuk menyampaikan secara akurat tujuan pesan dan informasi yang menyertainya. Pesan didefinisikan sebagai arahan yang diberikan kepada objek tertentu, biasanya menyerupai perintah.

2.2.12 Microsoft Azure

Azure adalah salah satu penyedia layanan *cloud* publik terkemuka. *Microsoft Azure* adalah *platform* komputasi awan yang membantu Anda membangun, *mendeploy*, dan mengelola aplikasi untuk mengatasi tantangan bisnis Anda. Perjalanan ini dimulai dengan dasar-dasar komputasi awan dan berlanjut dengan menjelaskan konfigurasi infrastruktur *Azure*, menjelajahi layanan *Azure*, dan bekerja pada *Virtual Memory System (VMS)* dan model *deployment* di *Azure*. Bab ini akan memperkenalkan komputasi awan kepada Anda dan menjelaskan evolusinya, jenis-jenis *model deployment* awan, dan layanan awan yang disediakan *Azure*. [53].

2.2.13 Black Box

Pengujian *black-box*, biasa disebut pengujian fungsional, adalah jenis pengujian yang mengabaikan cara kerja sistem atau komponen dan hanya berkonsentrasi pada keluaran yang dihasilkan sebagai respons terhadap masukan dan keadaan eksekusi tertentu. Biasanya, pendekatan pengujian *black-box* banyak digunakan dalam pengujian tingkat sistem dan skenario pengujian integrasi tertentu. Namun, ketika memasukkan komponen perangkat lunak pihak ketiga ke dalam sistem tanpa akses ke kode sumbernya, pengujian *black-box* perlu dilakukan untuk memverifikasi bahwa komponen ini memenuhi harapan pengguna [54].