

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

Landasan untuk memahami dan menjelaskan kejadian serta menyelesaikan masalah adalah teori dasar, atau mengembangkan pengetahuan dalam suatu area tertentu. Berikut yang termasuk teori dasar yaitu:

2.1.1 Pengertian Pengembangan

Tindakan menciptakan, merancang, atau mengembangkan ide atau produk yang sudah ada seperti bahan terbuka disebut pengembangan. Tujuan proses pengembangan ini adalah untuk memberikan keluaran yang lebih baik, bermanfaat, dan sesuai dengan tujuan yang dimaksudkan. Proses pengembangan dapat mencakup berbagai tahap, termasuk perencanaan, desain, implementasi, evaluasi, dan penyempurnaan. Penting untuk mencatat bahwa dalam konteks pendidikan atau pembelajaran, pengembangan bahan ajar adalah suatu aspek penting. Bahan ajar yang baik dan efektif dapat berkontribusi pada hasil pembelajaran yang lebih baik bagi siswa atau peserta pelatihan. Proses pengembangan bahan ajar biasanya mencakup penelitian, perancangan, pengujian, dan penyesuaian konten dan metode pembelajaran untuk mencapai tujuan pendidikan tertentu (Fajariadi et al., 2023).

2.1.2 Aplikasi

Aplikasi adalah program perangkat lunak yang diciptakan untuk menjalankan tugas atau perintah tertentu sesuai dengan tujuan pembuatannya. Aplikasi dapat mencakup berbagai jenis tugas dan fungsi, mulai dari aplikasi produktivitas hingga hiburan, dan dapat membantu dalam pemecahan masalah atau pemrosesan data

sesuai dengan kebutuhan pengguna. Aplikasi ini dapat dijalankan pada berbagai *platform*, termasuk perangkat *mobile*, komputer, dan perangkat lainnya. Mereka seringkali memanfaatkan berbagai teknik pemrosesan data untuk mencapai tujuan yang diinginkan, termasuk komputasi dan pemrosesan data yang sesuai. Dalam *era digital*, peran aplikasi dalam konteks teknologi *modern* sangat penting menyediakan alat dan solusi untuk berbagai keperluan, dan terus berkembang untuk memenuhi kebutuhan pengguna *modern* (Suwarno et al., 2023).

2.1.3 SDLC (*Software Development Life Cycle*)

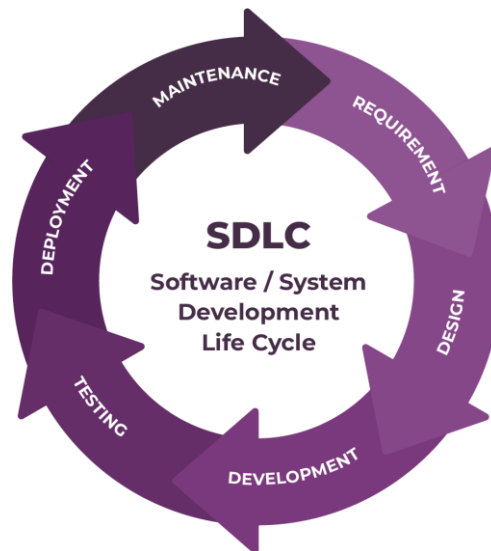
SDLC adalah singkatan dari *Software Development Life Cycle*, yang dalam bahasa Indonesia dikenal sebagai Siklus ini membantu memastikan bahwa setiap tahap pengembangan perangkat lunak dilakukan efektif dan sesuai dengan kebutuhan dan tujuan yang telah ditentukan untuk mengelola proses pengembangan dari awal hingga akhir. Metode SDLC adalah Cara di mana pengembangan perangkat lunak diorganisasi dan dijalankan selama seluruh siklus hidupnya. Metode SDLC sangat penting dalam pengembangan perangkat lunak karena membantu mengorganisasi dan mengelola proyek dengan efisien, memastikan kualitas, dan meminimalkan risiko kesalahan (Komalasari & Indaryono, 2021).

Berikut adalah langkah-langkah umum yang digunakan metode SDLC:

1. Perencanaan (*Planning*): Tahap ini melibatkan perencanaan awal proyek. Tim proyek mengidentifikasi tujuan proyek, kebutuhan pengguna, dan perencanaan umum. Ini juga melibatkan studi kelayakan, di mana Tim mengevaluasi apakah proyek ini layak dilaksanakan dari segi teknis, ekonomi, dan operasional.

2. Analisis (*Analysis*): Dalam tahap analisis, Tim proyek bekerja sama dengan pemangku kepentingan dan pengguna untuk mengidentifikasi kebutuhan bisnis yang akan diselesaikan oleh sistem yang akan dikembangkan. Ini termasuk pengumpulan dan dokumentasi persyaratan sistem.
3. Desain (*Design*): Pada tahap ini, perancangan sistem dilakukan berdasarkan persyaratan yang ditemukan pada tahap analisis. Ini mencakup merancang arsitektur sistem, antarmuka pengguna, *database*, dan detail teknis lainnya.
4. Pengembangan (*Development*): Di tahap ini, Tim proyek mulai mengimplementasikan desain yang telah dibuat sebelumnya. Ini melibatkan penulisan kode program, pengembangan *database*, integrasi komponen, dan pengujian unit.
5. Pengujian (*Testing*): Pengujian adalah tahap kunci dalam SDLC. Ini mencakup pengujian berbagai aspek sistem, termasuk pengujian fungsi, integrasi, keamanan dan kinerja. Tujuan pengujian adalah untuk memverifikasi bahwa sistem berfungsi sesuai yang diharapkan dan bebas dari cacat atau masalah.
6. Implementasi (*Deployment*): Setelah sistem telah diuji dan dianggap siap, ini dapat diimplementasikan ke lingkungan produksi. Ini mungkin melibatkan migrasi data, pelatihan pengguna, dan persiapan untuk penggunaan sehari-hari.

7. Pemeliharaan (*Maintenance*): Tahap pemeliharaan adalah fase lanjutan dalam siklus SDLC. Ini melibatkan pemeliharaan rutin sistem, penanganan perbaikan, perbaruan, dan perbaikan kecil. Pemeliharaan sistem ini diperlukan untuk menjaga sistem beroperasi secara efisien selama masa pakainya.
8. Evaluasi (*Evaluation*): Evaluasi sistem dapat dilakukan secara berkala guna memastikan sistem memenuhi tujuan bisnis dan keperluan pengguna. Evaluasi ini dapat mengarah pada perbaikan, pengembangan lebih lanjut, atau bahkan penggantian sistem jika diperlukan (Suryana & Yulianti, 2021).



Gambar 2.1 Metode SDLC
Sumber: (Binus academy, 2022)

2.1.4 UML (*Unified Modelling Language*)




UML atau *Unified Modeling Language* adalah suatu bentuk pemodelan yang diterapkan dalam rekayasa perangkat lunak untuk tujuan memvisualisasikan, menspesifikasikan, dan mendokumentasikan perancangan sistem berorientasi objek

atau yang dikenal sebagai OOP (*Object-Oriented Programming*). UML adalah bahasa grafis yang memungkinkan para pengembang untuk mengilustrasikan efektif dan sesuai dengan kebutuhan dan tujuan, menggunakan notasi-notasi grafis seperti diagram kelas, diagram sekuen, dan banyak lagi. UML dapat diterapkan pada berbagai tingkat abstraksi, dari tingkat konseptual hingga tingkat rincian implementasi. Ini memungkinkan pemodelan sistem perangkat lunak dari sudut pandang yang berbeda. UML dirancang untuk bersifat independen dari bahasa pemrograman dan teknologi tertentu, sehingga dapat digunakan dalam berbagai konteks pengembangan. UML adalah alat yang sangat penting dalam rekayasa perangkat lunak karena membantu para pengembang untuk berkomunikasi, merencanakan, dan mendokumentasikan perancangan perangkat lunak secara lebih sistematis dan terstruktur. Ini juga memfasilitasi pengembangan perangkat lunak berorientasi objek dan memungkinkan berbagai pemangku kepentingan untuk bekerja dengan bahasa yang sama untuk menggambarkan sistem (Dahoklory et al., n.d.).

1. *Use Case* Diagram

Kasus Pemanfaatan Di bawah *Unified Modeling Language* (UML), sejenis diagram yang disebut diagram *use case* digunakan untuk menggambarkan bagaimana sistem berinteraksi satu sama lain, umumnya perangkat lunak, dan pihak luar yang mengambil bagian dalam pertukaran. Diagram *use case* membantu memahami operasi utama sistem serta interaksi antara aktor dan sistem. Memanfaatkan diagram *use case* adalah merancang sistem dengan fokus pada fungsionalitas dan interaksi dengan pengguna dan entitas eksternal.

Tabel 2. 1 Simbol *usecase* diagram

Gambar	Keterangan
 actor	Aktor adalah tokoh yang mewakili kelompok. Simbol-simbol ini mewakili banyak pihak yang terkait dengan sistem, termasuk pengguna dan sistem tambahan yang terintegrasi ke dalam sistem utama.
 use case	<i>Use Case</i> adalah representasi grafis dari operasi sistem.
 usecase realization	Implementasi <i>use case</i> adalah representasi tindakan yang terjadi di dalam sistem dan eksklusif untuk sistem tersebut.
	Pekerja berfungsi sebagai simbol peran yang dimainkan perkerja dalam sistem.
 entity	Entitas adalah simbol yang menggambarkan entitas yang diproses oleh pekerja
	Relasi Simbol yang dikenal sebagai hubungan menghubungkan use case dengan aktor atau <i>use case</i> lainnya.






Sumber: (Era Azriani, 2023)

2. Activity Diagram

Diagram Aktivitas (*Activity Diagram*) merupakan satu jenis diagram dalam *Unified Modeling Language* (UML) yang bertujuan untuk menggambarkan alur kerja atau aktivitas dalam suatu sistem, proses bisnis, atau alur kerja tertentu. Diagram ini membantu dalam memvisualisasikan tindakan yang diperlukan untuk

menyelesaikan suatu tugas atau prosedur tertentu dan menunjukkan bagaimana aktivitas-aktivitas tersebut berinteraksi satu sama lain. *Activity Diagram* digunakan dalam berbagai konteks, termasuk untuk menjelaskan alur kerja sistem atau proses dalam rekayasa perangkat lunak, dalam bisnis untuk mendokumentasikan proses bisnis, dan dalam pengembangan sistem lainnya di mana pemahaman alur kerja dan aktivitas sangat penting.

Tabel 2. 2 Simbol *activity diagram*




Gambar	Keterangan
	<i>Start State</i> merupakan simbol yang menandakan dimulainya suatu kegiatan.
	<i>Stop State</i> / Keadaan berhenti merupakan simbol yang menandakan berakhirnya suatu kegiatan.
	<i>State</i> merupakan simbol yang digunakan untuk mewakili tindakan yang sedang berlangsung.
	Kondisi status memberikan isyarat visual tentang status operasi yang sedang berjalan.
	Simbol yang digunakan untuk mewakili peralihan dari satu keadaan ke keadaan lain disebut transisi keadaan .

Sumber: (Era Azriani, 2023)

3. *Sequence Diagram*

Dalam *Unified Modeling Language (UML)*, diagram *sequence*, disebut juga diagram urutan, adalah suatu bentuk diagram yang digunakan untuk menunjukkan bagaimana objek berinteraksi dalam suatu sistem dalam konteks berurutan atau urutan waktu tertentu. Diagram ini membantu dalam pemodelan dan visualisasi interaksi yang terjadi selama jangka waktu tertentu antar item dalam sistem.

Tabel 2. 3 Simbol *sequence* diagram



Gambar	Keterangan
 actor	Actor merupakan metafora bagi banyak pihak dalam sistem.
 	Objek merupakan Simbol mendefinisikan item dan interaksinya satu sama lain.
	Pesan objek merupakan Notasi yang menggambarkan pertukaran informasi antara dua objek disebut pesan objek.

Sumber: (Era Azriani, 2023)

4. Class Diagram

Unified Modeling Language (UML) mempunyai bentuk diagram yang disebut diagram kelas (*Class Diagram*) yang, dalam sistem berorientasi objek, digunakan untuk menggambarkan hierarki kelas. Item sistem dan hubungannya dimodelkan dengan bantuan diagram ini. Struktur sistem berorientasi objek dirancang menggunakan diagram kelas, yang menunjukkan kelas, properti, metode, dan hubungan yang diperlukan antara kelas-kelas ini.

Tabel 2. 4 Simbol *class* diagram

Gambar	Keterangan
	<p>Kelas adalah Sebuah kelas diwakili oleh sebuah simbol.</p>
	<p>Asosiasi merupakan Simbol “asosiasi” menunjukkan hubungan antar kelas..</p>

Sumber: (Era Azriani, 2023)

2.2 Teori Khusus

Teori khusus adalah hipotesis yang lebih terfokus dan mendalam dalam menjelaskan aspek-aspek tertentu dari suatu topik atau masalah penelitian. Berikut yang termasuk dalam teori khusus yaitu:

2.2.1 *Digital Library*

Perpustakaan *digital*, sering dikenal sebagai repositori *digital*, adalah sistem yang menyimpan dan mengelola koleksi berbagai jenis materi di dalam bentuk *file* elektronik (*digital*). Ini mencakup buku elektronik (*e-books*), jurnal ilmiah, gambar, audio, video, dokumen teks, dan berbagai jenis konten digital lainnya. *Digital library* mengadopsi menggunakan teknologi informasi dan komunikasi untuk mempermudah akses dan distribusi konten ini melalui jaringan komputer, biasanya melalui internet atau intranet. Perpustakaan adalah tempat atau koleksi yang menyimpan berbagai jenis informasi dan bahan bacaan yang mencakup ilmu pengetahuan, hiburan, rekreasi, dan bahkan bahan yang berkaitan dengan aspek spiritual dan ibadah. Ini adalah sumber daya penting bagi masyarakat untuk memenuhi berbagai kebutuhan informasi dan pengetahuan mereka. Keuntungan

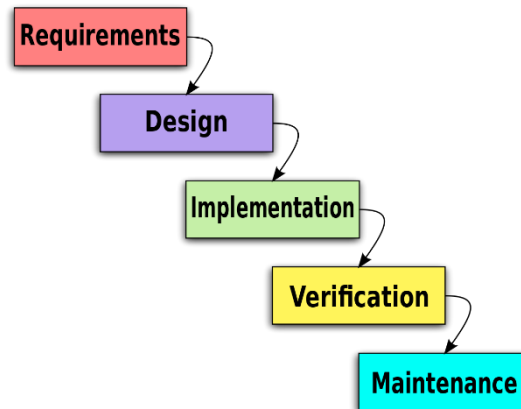
dari *digital library* meliputi kemudahan akses, pencarian yang lebih efisien, dan kemampuan untuk mengakses sumber daya ini dari berbagai lokasi dan perangkat yang terhubung ke internet. Ini juga membuka peluang untuk berbagi pengetahuan dan sumber daya dengan lebih banyak orang di seluruh dunia. *Digital library* memiliki peran penting dalam pendidikan, penelitian, dan penyediaan informasi di *era digital* (Yuda & Muludi, 2021).

2.2.2 Waterfall

Metode air terjun merupakan paradigma pengembangan perangkat lunak yang sering diterapkan dalam rekayasa perangkat lunak. Paradigma Sekuensi *Linier* atau Model Urutan *Linier* adalah istilah umum yang digunakan untuk menggambarkan paradigma *Waterfall*. Karena pendekatannya metadis dan berurutan dalam pembuatan perangkat lunak. Proses pengembangan mengharuskan setiap langkah diselesaikan sebelum melanjutkan ke langkah berikutnya. Nama "*Waterfall*" menggambarkan cara model ini berfungsi. Setiap tahap dalam proses pengembangan mengalir seperti air terjun, dengan satu tahap mengikuti tahap sebelumnya. Tahap yang paling atas adalah analisis kebutuhan, diikuti oleh perencanaan, desain, implementasi, pengujian, integrasi, dan pemeliharaan. Model ini mensyaratkan bahwa Penyelesaian sempurna setiap langkah diperlukan sebelum melanjutkan ke langkah berikutnya. Artinya, tidak ada perubahan atau revisi yang diizinkan di tahap sebelumnya setelah tahap tersebut selesai. Ini membuat model ini kurang fleksibel dalam menghadapi perubahan kebutuhan atau perbaikan yang diperlukan (Fajariadi et al., 2023).

Berikut tahapan atau langkah-langkah metode *waterfall* yaitu:

1. *Requirement/Perencanaan*: Tahap ini melibatkan komunikasi dengan klien untuk memahami kebutuhan mereka. Data kebutuhan diperoleh melalui wawancara, diskusi, atau survei. Hasil dari tahap ini adalah pemahaman tentang apa yang diinginkan oleh pelanggan dan keterbatasan saat mengembangkan perangkat lunak.
2. *Design System (Desain Sistem)*: Langkah ini mencakup pendefinisian apa yang telah diketahui pada tahap sebelumnya dan merupakan kelanjutan dari tahap perencanaan. Penentuan arsitektur sistem secara keseluruhan dibantu oleh desain sistem.
3. *Pengkodean dan Tes*: Pada tahap ini, aplikasi sebenarnya dikembangkan berdasarkan desain sistem. Pengembang akan mengimplementasikan apa yang telah dipelajari dan didesain dalam tahap sebelumnya untuk membuat aplikasi yang memenuhi permintaan pengguna.
4. *Integrasi dan Sistem Tes*: Tahap ini melibatkan pengujian dan pengujian sistem. Aplikasi yang telah dibangun akan diuji untuk menjamin bahwa mereka memenuhi permintaan pengguna dan beroperasi sebagaimana dimaksud. Tahap ini adalah tahap akhir dalam pengembangan perangkat lunak sebelum mengirimkan produk ke klien.
5. *Operasi dan Pemeliharaan Sistem*: Setelah pengguna menerima *gadget*, tahap ini melibatkan pemeliharaan sistem. Ini mencakup perbaikan kesalahan yang mungkin muncul setelah perangkat lunak digunakan oleh pengguna (Ilham et al., 2020).



Gambar 2. 2 Metode *waterfall*

Sumber: (medium, 2020)

2.2.3 *Android*

Android adalah dari sistem operasi yang paling sering digunakan di seluruh dunia dan digunakan oleh miliaran orang di berbagai perangkat di seluruh dunia. Sistem operasi *Android* didasarkan pada *kernel Linux*. *Kernel Linux* adalah inti dari sistem operasi manajemen sumber daya perangkat keras seperti prosesor, memori, dan perangkat *input/output*. *Android* adalah proyek sumber terbuka, yang berarti bahwa kode sumbernya tersedia untuk umum. Hal ini memungkinkan pengembang dan produsen perangkat untuk mengakses, memodifikasi, dan mendistribusikan *Android* sesuai dengan kebutuhan mereka. *Android* dirancang untuk berbagai jenis perangkat, termasuk ponsel pintar (*smartphone*) dan komputer *tablet*. Ini menciptakan ekosistem perangkat yang luas yang berjalan pada sistem operasi yang sama. *Android* menyediakan alat dan lingkungan pengembangan yang kuat, seperti *Android Studio*, untuk pengembang aplikasi. Hal ini memudahkan pengembangan dan distribusi aplikasi yang beragam, mulai dari aplikasi produktivitas hingga permainan. *Gmail*, *Google Maps*, *Google Drive*, dan masih banyak lagi layanan

Google terintegrasi erat dengan *Android*. Ini memudahkan pengguna untuk mengakses dan mengelola layanan tersebut melalui perangkat *Android* (Nugraha & Munawar, 2022).



Gambar 2. 3 Logo *android*
Sumber: (tek.id, 2020)

Adapun perkembangan versi *android*, Tabel berikut menunjukkan evolusi versi *Android*:

Tabel 2. 5 Tabel versi *android*

No	Versi	Nama	Tahun
1	<i>Android 1.0</i>	<i>Apple pie (Alpha)</i>	2008
2	<i>Android 1.1</i>	<i>Banana bread (Beta)</i>	2009
3	<i>Android 1.5</i>	<i>Cupcake</i>	2009
4	<i>Android 1.6</i>	<i>Donut</i>	2009
5	<i>Android 2.0</i>	<i>Éclair</i>	2009
6	<i>Android 2.2</i>	<i>Froyo (Frozen Yogurt)</i>	2010
7	<i>Android 2.3</i>	<i>Gingerbread</i>	2010
8	<i>Android 3.0</i>	<i>Honeycomb</i>	2010
9	<i>Android 4.0</i>	<i>Ice cream sandwich</i>	2011

Tabel 2.5 (Lanjutan)

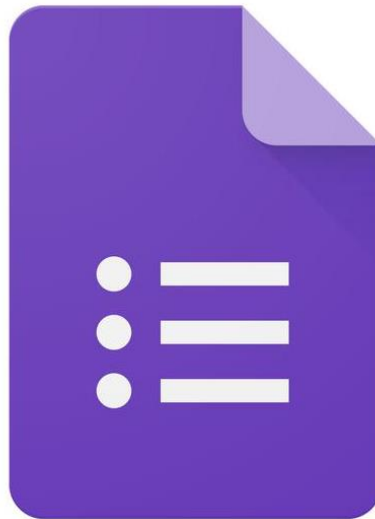
10	<i>Android 4.1</i>	<i>Jelly Bean</i>	2012
11	<i>Android 4.4</i>	<i>Kitkat</i>	2013
12	<i>Android 5.0</i>	<i>Lollipop</i>	2014
13	<i>Android 6.0</i>	<i>Marshmallow</i>	2016
14	<i>Android 7.0</i>	<i>Nougat</i>	2016
15	<i>Android 8.0</i>	<i>Oreo</i>	2017
16	<i>Android 9.0</i>	<i>Pie</i>	2018
17	<i>Android 10</i>	<i>Quince Tart</i>	2019
18	<i>Android 11</i>	<i>Red Velvet Cake</i>	2020
19	<i>Android 12</i>	<i>Snow Cone</i>	2021
20	<i>Android 13</i>	<i>Tiramisu</i>	2022

Sumber: (tika, 2023)

2.2.4 Google Form

Layanan *Google* Formulir memungkinkan pengguna dengan cepat dan mudah membuat formulir atau *kuesioner online* dengan tujuan mengumpulkan data. *Google Form* memungkinkan pengguna untuk membuat formulir yang dapat berisi berbagai jenis pertanyaan, seperti pertanyaan teks, pilihan ganda, isian singkat, dan banyak lagi. Pengguna dapat merancang formulir sesuai dengan kebutuhan mereka. Formulir yang dibuat dengan *Google Form* disimpan di *Google Drive* pengguna. Ini memungkinkan pengguna untuk dengan mudah mengakses, mengedit, dan mengelola formulir mereka. Antarmuka *Google Form* dirancang agar mudah digunakan. Pengguna tidak perlu memiliki pengetahuan teknis khusus untuk

membuat formulir. Semua proses pembuatan dan pengelolaan formulir dilakukan melalui antarmuka web yang intuitif (Astuti, 2021).



Gambar 2. 4 Logo *google form*
Sumber: (UMSU, 2021)

2.2.5 *Google Sheet*

Google Sheets adalah aplikasi *spreadsheet* milik Lembar kerja dapat dibuat, diedit, dan dikelola secara online menggunakan *Google*. Dengan *Google Sheets*, Anda dapat melakukan banyak fungsi yang umumnya terkait dengan *spreadsheet*, seperti membuat tabel, menghitung rumus, membuat grafik, berbagi dan Bekerja sama secara *real time* dengan orang lain, serta menyimpan data secara aman di *cloud*. Keuntungan utama dari *Google Sheets* adalah kemampuannya untuk berkolaborasi dengan mudah. Beberapa orang dapat bekerja pada lembar kerja yang sama secara bersamaan, mengomentari sel, dan melihat perubahan dalam waktu nyata. Selain itu, data yang disimpan di *Google Sheets* dapat diakses melalui berbagai perangkat yang terkoneksi internet, sehingga Anda tidak harus khawatir kehilangan data jika perangkat Anda rusak atau hilang. *Google Sheets* juga

menyediakan berbagai fungsi *spreadsheet* yang kuat, seperti rumus, makro otomatisasi, pemformatan yang kaya, dan integrasi dengan aplikasi *Google* lainnya, seperti *Google Forms* untuk survei dan *Google Data Studio* untuk visualisasi data. Selain itu, *Google Sheets* tersedia secara gratis dengan akun *Google*, oleh karena itu Anda tidak perlu membayar lebih untuk menggunakannya (Hidayat et al., 2021).



Gambar 2. 5 Logo *google sheets*
Sumber: (*google*, 2020)

2.2.6 Adobe Animate

Adobe Animate adalah aplikasi pengembangan dari *Adobe Flash Professional* yang menyediakan berbagai fitur canggih untuk mendesain grafik vektor dan animasi. *Adobe Animate* adalah salah satu alat yang kuat dalam industri animasi dan pembuatan konten multimedia. Dengan fokus pada teknologi HTML5, ia mengikuti tren perkembangan web modern dan memungkinkan pengembang untuk menciptakan animasi dan konten interaktif yang responsif dan beragam *Adobe Animate* sebelumnya dikenal sebagai *Adobe Flash Professional CC*. Ini merupakan perkembangan dari versi sebelumnya, termasuk *Adobe Flash CS6*. Seiring

berjalannya waktu, *Adobe* mengubah nama produk ini menjadi *Adobe Animate* untuk mencerminkan fokusnya pada animasi dan konten berbasis HTML5. *Adobe Animate* digunakan untuk membuat animasi komputer dan konten multimedia. Ini mencakup desain grafis vektor, pembuatan animasi 2D, dan pembuatan konten interaktif. *Adobe Animate* memungkinkan konten yang dihasilkan dipublikasikan pada berbagai *platform* dan media, termasuk televisi, video *online*, situs *web*, dan video *game*. Ini membuatnya sangat berguna dalam industri hiburan, pendidikan, periklanan, dan banyak lagi. *Adobe Animate* mendukung pengembangan konten berbasis HTML5. Dengan HTML5, animasi dan konten interaktif dapat dijalankan tanpa ketergantungan pada pemutarnya dan kompatibel dengan berbagai perangkat dan peramban *web*. *Adobe Animate* sering diidentikkan dengan pembuatan animasi, dan digunakan oleh banyak profesional animasi untuk menciptakan animasi 2D yang berkualitas tinggi (Rahmawati et al., 2023).



Gambar 2. 6 Logo *adobe animate*
Sumber: (Wikipedia, 2020)

2.3 Penelitian Terdahulu

1. (Sianturi & Hendriani, 2021) **Perancangan Sistem *Library* Berbasis Web Menggunakan Metode *Waterfall***. ISSN 2338-1523 E-ISSN 2541-576X, Salah satu sumber informasi yang perlu dimiliki sekolah dalam rangka memajukan ilmu pengetahuan dan menyediakan sarana prasarana baik bagi siswa maupun masyarakat luas adalah perpustakaan. Untuk membantu siswa menemukan informasi tentang buku-buku yang ingin mereka baca dan pinjamkan, serta untuk membantu dan mempercepat tenaga administrasi dalam menyusun laporan data buku-buku yang dipinjam dan dikembalikan, proyek ini bertujuan untuk membangun sistem perpustakaan berbasis web. Model air terjun, suatu pendekatan metodologis dalam pengembangan sistem yang mencakup tahapan perencanaan, analisis, desain, dan implementasi, adalah pendekatan yang digunakan dalam penelitian ini.
2. (Mailasari, 2019) **Sistem Informasi Perpustakaan Menggunakan Metode *Waterfall***. ISSN: 2301-7988 E-ISSN : 2581-0588. Metode konvensional masih digunakan pada sistem perpustakaan SMP Negeri 3 Kota Bekasi. Misalnya pencarian, peminjaman dan pengembalian masih menggunakan cara pencarian manual, dan sistem peminjaman dan pengembalian masih menggunakan sistem tulisan tangan pada buku besar. Akibatnya pelayanan perpustakaan menjadi kurang efektif dan efisien. Permasalahan tersebut memerlukan pengembangan sistem dengan menggunakan pendekatan penelitian air terjun, dengan prosedur yang disusun untuk memastikan keluaran sistem berkualitas tinggi. Luaran penelitian ini berupa sistem

informasi perpustakaan berbasis web yang dibangun dengan *database* MySQL dan bahasa pemrograman PHP. Pengujian *black box* akan digunakan untuk mengevaluasi temuan penyelidikan ini.

3. (Ilham et al., 2020) **Pengembangan Sistem Perpustakaan Universitas Muhammadiyah Malang Berbasis *Android* Dengan Metode *Waterfall***. ISSN: 2714-7975 E-ISSN: 2716-1382. Setiap institusi pendidikan mempunyai perpustakaan sebagai sumber informasi dan sumber pendukung, namun perpustakaan perguruan tinggi dan universitas sangat penting untuk perdebatan. Berdasarkan jajak pendapat yang dilakukan Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) baru-baru ini, terdapat peningkatan signifikan dalam penggunaan perangkat seluler. Terdapat lonjakan signifikan dalam jumlah pengguna internet di Indonesia antara tahun 2013 dan 2017, dengan jumlah pengguna sebesar 61,26 juta orang pada saat itu. Perilaku pengguna internet sebagai berikut, bergantung pada gadget yang mereka gunakan untuk mengakses internet: Sebanyak 44,16% konsumen menggunakan perangkat seluler, 4,49% menggabungkan laptop dan perangkat seluler, dan 39,28% menggunakan PC dan perangkat seluler. Hal ini menunjukkan bahwa perilaku pengguna internet Indonesia cukup lazim dalam hal penggunaan.
4. (Anaz & Sari, 2023) **Rancang Bangun Aplikasi Perpustakaan Pada SMP Negeri 39 Kota Bekasi Berbasis *Android***. E-ISSN: 2715-8160. SMP Negeri 39 masih menerapkan prosedur manual yang dicatat dalam buku agenda untuk mengolah data buku perpustakaan, sehingga menyebabkan seringnya

terjadi kesalahan dalam laporan buku dan mencari buku, yang memakan waktu. Perpustakaan saat ini harus mampu memberikan pengunjung akses terhadap informasi yang cepat, akurat, dan bermanfaat. Tujuan dari adanya aplikasi perpustakaan berbasis *mobile* adalah untuk mempercepat dan meningkatkan pelayanan yang diberikan kepada pengunjung perpustakaan, sehingga akan membantu dalam proses belajar mengajar. Sistem ini dibuat menggunakan *Android Studio* dan aplikasi *mobile* bergaya air terjun. Sistem kerja perpustakaan terintegrasi dalam pelayanannya melalui pemanfaatan teknologi.

5. (Bramantoro, 2020) ***Stateful library service system design and implementation in Saudi Arabia***. ISSN: 2088-8708. Penelitian ini bertujuan untuk mengembangkan dan mempublikasikan layanan yang mampu membaca semua data yang diperlukan dari sistem manajemen perpustakaan, dengan fokus tambahan pada peningkatan layanan melalui penerapan teknik kewarganegaraan. Tantangan dalam sistem perpustakaan terkait khususnya dengan jumlah pengunjung, layanan, kapasitas penyimpanan, dan ukuran organisasi.
6. (Trisnadoli & Yanti, 2022) ***ANALYSIS OF PROTOTYPING METHODS IMPLEMENTATION FOR LIBRARY INFORMATION SYSTEMS DEVELOPMENT IN SMAN 3 BENGKALIS***. ISSN 2089-8673 (*Print*) | ISSN 2548-4265 (*Online*). Perpustakaan SMAN 3 Bengkalis menghadapi beberapa tantangan, termasuk sistem konvensional yang rentan terhadap kerusakan fisik, pendaftaran anggota menggunakan kartu kertas, dan

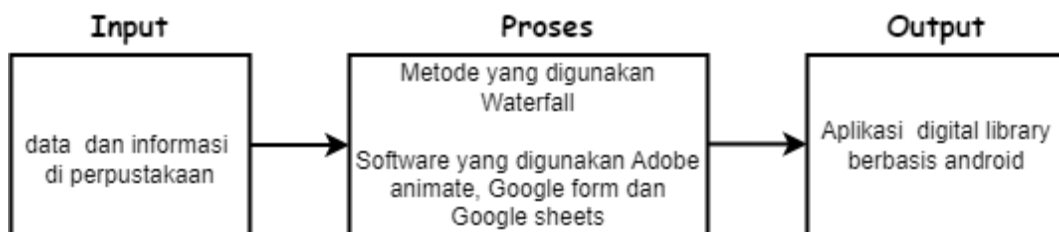
kesulitan dalam pengadaan dan pengelolaan buku. Untuk mengatasi masalah ini, penelitian ini mengusulkan pembuatan sistem informasi online dengan menggunakan *prototyping*. Evaluasi dilakukan melalui UAT (*User Acceptance Testing*) dan pengujian ketergunaan dengan standar ISO 9126 dan 9241-111.

7. (Saputro et al., 2020) ***Development of a Web-Based Online Library Information System at the National University Using the Waterfall Method.*** *E-ISSN 2685-4236*. Sesuatu yang diasosiasikan dengan Pustaka atau perpustakaan adalah: sedikit sumber bagan dari sekolah sampai-sampai pengetahuan tidak berasal dari informasi atau pengetahuan dari perpustakaan. Sehubungan dengan kemajuan teknologi ini, manusia kini dapat bekerja lebih efektif dan efisien di tempat kerja. Misalnya: membuat sistem konvensional yang lebih terkomputerisasi. Pemanfaatan fitur *online* pada *website* juga dapat meningkatkan efisiensi dan efektivitas dalam pembukuan dan pemrosesan transaksi. Penelitian ini menggunakan sistem informasi *website* berbasis PHP. Berdasarkan hasil penelitian, penulis mengimplementasikan hasil penelitian tersebut pada bidang sebagai berikut: Pengembangan Sistem Informasi *Online* Sekolah Tinggi Nasional Dengan Metodologi *Waterfall*.

2.4 Kerangka Pemikiran

Penelitian untuk menciptakan aplikasi yang membantu mempercepat proses atau sistem informasi di perpustakaan memiliki sejumlah tujuan dan manfaat yang jelas. Tujuan utama adalah meningkatkan efisiensi dan produktivitas staf perpustakaan. Dengan program ini, Anda dapat menyelesaikan lebih banyak

pekerjaan dalam waktu lebih singkat. Aplikasi ini dibuat untuk mengotomatisasi banyak tugas yang dilakukan dalam perpustakaan, termasuk pencarian buku, pemrosesan pinjaman, dan pengelolaan data buku. Memungkinkan anggota perpustakaan untuk mencari dan meminjam buku dengan lebih mudah. Aplikasi ini dapat memungkinkan perpustakaan untuk melacak dan melaporkan data penggunaan buku, statistik pinjaman, dan sebagainya.



Gambar 2. 7 Kerangka Pemikiran
Sumber: (Penelitian 2024)

Demikian penjelasannya berdasarkan gambar di atas:

1. *Input*: memasukkan Identifikasi kebutuhan dasar sistem perpustakaan, termasuk manajemen data buku dan peminjaman.
2. *Proses*: Merancang tampilan antarmuka aplikasi *digital library* menggunakan Software seperti *Adobe Animate*, *Google Form*, dan *Google Sheet*, berkontribusi pada pengembangan aplikasi dan manajemen data peminjaman buku. Ini mencakup desain tata letak halaman, ikon, tombol, dan elemen antarmuka lainnya. Merancang formulir peminjaman buku dan basis data yang akan digunakan di *Google Form* dan *Google Sheet*. Membuat formulir peminjaman buku menggunakan *Google Form* dan menghubungkannya dengan *Google Sheet* untuk menyimpan data peminjaman.

3. *Output:* hasil akhir Aplikasi perpustakaan *digital* yang disiapkan untuk digunakan oleh perpustakaan, yang mencakup manajemen data buku dan informasi peminjam. Melakukan pengujian aplikasi untuk memastikan kualitas dan kehandalan aplikasi dan Memastikan bahwa data buku dan data peminjam tersimpan dengan benar di *Google Sheet*.