

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Berikut ini merupakan sebuah analisis terhadap berbagai penelitian terdahulu yang telah berkontribusi pada penelitian ini

Tabel 2. 1 Tabel Penelitian Terdahulu

Penelitian Terdahulu		
1	Penulis & Tahun	Prasetyo et al., 2021[6]
	Judul	<i>Development of Android-Based Public Transportation Application Using SDLC Waterfall Method</i>
	Permasalahan	Informasi jadwal tidak terpusat dan belum berbasis aplikasi <i>mobile</i>
	Hasil	Aplikasi berbasis Android dengan fitur jadwal terstruktur
	Gap Analisis	Tidak mengintegrasikan rute & penyimpanan favorit
	Faktor Kritis	Akses informasi jadwal
	Kontribusi Penelitian Ini	Integrasi fitur pencarian rute dan penyimpanan jadwal
2	Penulis & Tahun	Ramadhan & Sari, 2020 [7]
	Judul	<i>Design of Public Transport Scheduling System in Urban Area Using Mobile Application</i>
	Permasalahan	Belum ada sistem digital terintegrasi untuk penjadwalan
	Hasil	Prototipe sistem penjadwalan dasar
	Gap Analisis	Tidak ada pengujian sistem ke pengguna
	Faktor Kritis	Kebutuhan jadwal harian pengguna

Penelitian Terdahulu		
	Kontribusi Penelitian Ini	Menyediakan aplikasi siap pakai dengan uji fungsional dan UAT
3	Penulis & Tahun	Widodo & Yusuf, 2022[8]
	Judul	<i>Integrating Real-Time Scheduling for Public Transportation in Smart City Applications</i>
	Permasalahan	Informasi <i>real-time</i> sulit dijangkau oleh masyarakat
	Hasil	Sistem <i>real-time</i> berbasis <i>dashboard</i>
	Gap Analisis	Tidak membahas personalisasi atau fitur favorit
	Faktor Kritis	<i>Realtime schedule</i> & estimasi
	Kontribusi Penelitian Ini	Penjadwalan berdasarkan input pengguna & penyimpanan preferensi
4	Penulis & Tahun	Pradana et al., 2023[9]
	Judul	<i>Mobile-Based Route and Schedule Planning System for Public Transportation in Jakarta</i>
	Permasalahan	Rute tidak optimal dan data sulit diakses
	Hasil	Aplikasi berbasis rute & waktu tempuh
	Gap Analisis	Belum ada fitur personalisasi atau UAT
	Faktor Kritis	Rekomendasi rute terbaik
	Kontribusi Penelitian Ini	Tambahkan fitur berbasis titik awal & tujuan pengguna
5	Penulis & Tahun	Zulkarnain & Sitorus, 2023[10]
	Judul	<i>Analysis of User Needs in Public Transportation Scheduling Applications</i>
	Permasalahan	Kebutuhan pengguna belum diteliti
	Hasil	Identifikasi fitur penting melalui studi kualitatif
	Gap Analisis	Tidak dilakukan pengembangan sistem
	Faktor Kritis	Preferensi fitur pengguna

Penelitian Terdahulu		
	Kontribusi Penelitian Ini	Implementasi langsung dari kebutuhan yang teridentifikasi
6	Penulis & Tahun	Maulana & Putri, 2022[11]
	Judul	<i>Design of Smart Scheduling System for Urban Bus in Surabaya</i>
	Permasalahan	Penjadwalan masih dilakukan manual
	Hasil	Desain sistem penjadwalan bus berbasis Android
	Gap Analisis	Tidak mencakup tahap implementasi penuh
	Faktor Kritis	Sistem cerdas berbasis mobilitas
	Kontribusi Penelitian Ini	Realisasi sistem dalam bentuk aplikasi jadi + pengujian
7	Penulis & Tahun	Natalia, 2023[12]
	Judul	Sistem Informasi Rute Transportasi Umum Berbasis Android
	Permasalahan	Akses rute dan trayek KRL belum terdigitalisasi
	Hasil	Aplikasi menampilkan rute KRL statis
	Gap Analisis	Belum ada fitur penjadwalan & tidak dilakukan uji pengguna
	Faktor Kritis	Digitalisasi trayek
	Kontribusi Penelitian Ini	Tambahkan fitur interaktif & evaluasi pengguna
8	Penulis & Tahun	Siregar, 2022[13]
	Judul	Penerapan Metode <i>Waterfall</i> dalam Aplikasi Informasi Transportasi Umum
	Permasalahan	Dokumentasi pengembangan sistem tidak lengkap
	Hasil	Penerapan tahapan <i>SDLC Waterfall</i>
	Gap Analisis	Tidak ada pengujian akhir dari sistem

Penelitian Terdahulu		
	Faktor Kritis	Proses sistematis
	Kontribusi Penelitian Ini	Tambahkan fase UAT dan dokumentasi berbasis praktik nyata
9	Penulis & Tahun	Wijaya, 2021[14]
	Judul	Perancangan Aplikasi Informasi Transportasi Umum di Yogyakarta
	Permasalahan	Informasi transportasi tersebar dan tidak konsisten
	Hasil	Informasi transportasi tersebar dan tidak konsisten
	Gap Analisis	Tidak mengelola fitur rute atau waktu spesifik
	Faktor Kritis	Centralisasi info transportasi
	Kontribusi Penelitian Ini	Tambahkan fungsi penjadwalan aktual dan filter lokasi
10	Penulis & Tahun	Wulandari, 2020[15]
	Judul	Sistem Penjadwalan Armada Transportasi dengan Antarmuka Mobile
	Permasalahan	Pengelolaan jadwal armada tidak efisien
	Hasil	Sistem visualisasi jadwal armada
	Gap Analisis	Fokus ke operator, bukan pengguna transportasi umum
	Faktor Kritis	Penjadwalan backend
	Kontribusi Penelitian Ini	Aplikasi untuk pengguna akhir (penumpang), bukan operator

Berbagai penelitian terdahulu telah dilakukan untuk menjawab kebutuhan digitalisasi informasi terhadap transportasi umum, khususnya dengan adanya pengembangan aplikasi berbasis Android. Studi-studi tersebut melakukan adopsi pendekatan metodologis seperti *SDLC Waterfall* [6][9][11][13], model

iteratif[8], serta menggunakan metode *prototyping* dan RAD[7][12][15], dan pada umumnya menekankan pentingnya dalam melakukan pengajian informasi rute dan jadwal perjalanan secara digital. Hasilnya akan menunjukkan bahwa penerapan teknologi mobile dapat meningkatkan efisiensi penyampaian informasi dan aksesibilitas bagi pengguna transportasi umum.

Meskipun demikian, kajian terhadap sepuluh penelitian tersebut mengungkapkan adanya kesenjangan baik dari sisi fungsionalitas sistem, pendekatan metodologi, maupun aspek pengguna. Banyak dari penelitian yang hanya sampai pada tahap perancangan dan simulasi, tanpa adanya melibatkan proses implementasi dan melakukan validasi sistem di lingkungan nyata [7][8][12]. Hal ini dapat menyebabkan adanya keterbatasan terhadap data empiris yang dapat menunjukkan keberhasilan sistem dalam memenuhi kebutuhan pengguna akhir secara langsung. Kurangnya pengujian sistem seperti *black box testing* atau *user acceptance test* mengindikasikan bahwa sistem yang telah dikembangkan belum dilakukan pengujian secara menyeluruh dari sisi keandalan dan kenyamanan pengguna.

Selain itu, sebagian besar dari sistem yang telah dilakukan pengembangan hanya menyajikan sebuah data secara statis tanpa menghadirkan fitur interaktif yang bersifat adaptif terhadap kebutuhan pengguna harian. Fitur seperti pencarian rute berbasis lokasi pengguna, penjadwalan perjalanan personal, dan penyimpanan jadwal favorit masih menjadi sebuah fitur yang jarang ditemui dalam sistem yang telah dibangun [9][10][14]. Padahal, studi tentang preferensi pengguna menunjukkan bahwa fitur-fitur ini akan sangat dibutuhkan untuk dapat meningkatkan kenyamanan dan efisiensi perjalanan, terutama di wilayah urban dengan banyak pilihan moda transportasi.

Dari aspek cakupan geografis, sebagian besar studi berfokus pada kota-kota besar seperti Jakarta dan Surabaya [9][11], atau tidak menyebutkan secara spesifik wilayah untuk dilakukan pengujian [6][7]. Akibatnya, konteks lokal yang mungkin sangat berpengaruh terhadap perilaku pengguna, kondisi infrastruktur transportasi, serta integrasi antar moda menjadi terabaikan. Hal ini

dapat menjadi sebuah peluang yang besar untuk dapat menghadirkan penelitian yang berfokus pada wilayah urban lain seperti Kota Tangerang, yang memiliki karakteristik mobilitas yang tinggi namun belum banyak disentuh oleh penelitian yang serupa.

Kelemahan lain yang tampak adalah bahwa meskipun SDLC *Waterfall* sering disebut sebagai sebuah metode yang digunakan, tidak semua penelitian menjalankan semua fase SDLC secara lengkap dan terdokumentasi. Banyak studi hanya menjelaskan tahap analisis dan desain, tetapi tidak mencantumkan tahapan implementasi dan evaluasi sistem secara sistematis[11][13]. Padahal, kekuatan utama dari metode SDLC *Waterfall* terletak pada urutan proses yang terstruktur dan dokumentasi yang komprehensif untuk setiap fase yang berada pada pengembangan sistem.

Oleh karena itu, penelitian ini akan mengambil posisi unik dengan cara mengembangkan aplikasi *scheduling* transportasi umum berbasis *Android* yang dirancang secara khusus untuk pengguna di wilayah Kota Tangerang. Aplikasi ini tidak hanya akan menyediakan informasi statis seperti jadwal maupun rute, tetapi juga akan mengintegrasikan fitur pencarian rute berdasarkan titik asal dan tujuan pengguna, penyimpanan rute favorit, serta penyajian antarmuka yang intuitif dan ramah pengguna. Pengembangan aplikasi ini akan dilakukan dengan menggunakan pendekatan SDLC *Waterfall* yang dijalankan secara menyeluruh, mulai dari analisis kebutuhan hingga akan melakukan pengujian teknis menggunakan metode *blackbox* dan validasi penerimaan pengguna melalui *user acceptance test*.

Kontribusi penelitian ini tidak hanya akan terletak pada pengembangan teknis sistem, tetapi juga pada aspek praktis dalam memberikan solusi nyata yang disesuaikan dengan kebutuhan lokal masyarakat perkotaan. Dengan menutup celah pada dimensi geografis, metodologis, dan fungsional yang belum ditangani oleh studi terdahulu, penelitian ini akan diharapkan menjadi sebuah fondasi kuat untuk melakukan pengembangan aplikasi transportasi umu

digital di masa mendatang yang lebih kontekstual, aplikatif, dan adaptif terhadap kebutuhan pengguna sebenarnya.

2.2 Teori tentang Topik Skripsi

2.2.1 Transportasi Umum di Perkotaan

Transportasi umum merupakan sebuah sistem layanan mobilitas yang dirancang untuk digunakan secara kolektif oleh masyarakat dalam skala yang besar. Moda transportasi umum seperti bus, kereta, dan angkot disediakan untuk mengakomodasi kebutuhan pergerakan penduduk secara efisien dan terjadwal. Keberadaan sistem ini akan menjadi sebuah penunjang vital aktivitas masyarakat, terutama di kawasan perkotaan yang padat dan memiliki tingkat mobilitas tinggi[16].

Dalam konteks Indonesia, khususnya di kota penyangga seperti Kota Tangerang, masih banyak tantangan dalam penerapan transportasi umum yang optimal. Permasalahan yang sering muncul adalah kurangnya informasi *real-time* terkait jadwal dan ketersediaan kendaraan, ketidakterpaduan antar moda, serta ketidakpastian waktu tempuh. Ketiga faktor ini berkontribusi terhadap rendahnya kepercayaan publik terhadap transportasi umum, sehingga pengguna lebih memilih untuk menggunakan kendaraan pribadi[17].

Penelitian ini berusaha menjawab tantangan tersebut dengan menyediakan informasi transportasi melalui aplikasi berbasis Android yang dapat menampilkan jadwal dan rute perjalanan secara jelas dan terorganisasi. Penguatan akses terhadap informasi ini diharapkan dapat meningkatkan kenyamanan serta efisiensi perjalanan pengguna transportasi umum di Kota Tangerang.

2.2.2 Sistem Penjadwalan Transportasi

Sistem penjadwalan dalam transportasi merupakan mekanisme untuk menentukan waktu keberangkatan, rute, dan durasi perjalanan secara sistematis dan terorganisir. Tujuannya adalah memastikan efisiensi operasional, keterpaduan moda, dan kepastian pelayanan bagi

pengguna. Sistem penjadwalan umumnya digunakan baik untuk kebutuhan operator dalam manajemen armada maupun untuk pengguna dalam merencanakan perjalanan[18].

Penjadwalan yang baik dapat membantu pengguna memilih moda yang sesuai dengan waktu dan rute terbaik. Selain itu, sistem ini mampu mengurangi waktu tunggu dan meminimalkan ketidakpastian, terutama dalam transportasi publik yang mengandalkan waktu tempuh sebagai faktor utama kenyamanan. Dalam banyak studi, aplikasi penjadwalan terbukti meningkatkan keteraturan dan keterjangkauan informasi bagi masyarakat[19].

Dengan menerapkan sistem penjadwalan ke dalam aplikasi mobile, penelitian ini bertujuan memberikan solusi digital yang mampu menyajikan rute dan jadwal secara efisien. Pengguna dapat melihat jadwal transportasi secara *real-time* maupun berbasis waktu tetap, sehingga mampu merancang perjalanan dengan lebih akurat.

2.2.3 Aplikasi *Mobile* sebagai Solusi Layanan Publik

Aplikasi *mobile* telah berkembang menjadi media layanan publik yang cepat, personal, dan efisien. Perangkat lunak yang dijalankan pada *smartphone* atau tablet ini dirancang untuk memberikan kemudahan akses dan pengalaman pengguna yang lebih responsif dibandingkan layanan berbasis *desktop*. Dengan jumlah pengguna *Android* yang dominan di Indonesia, platform *mobile* menjadi sarana utama untuk penyampaian informasi publik secara langsung[20].

Dalam sektor transportasi, aplikasi *mobile* memainkan peran penting dalam menyediakan data real-time terkait jadwal, rute, lokasi kendaraan, hingga estimasi waktu tempuh. Keunggulan ini menjadikan aplikasi sebagai sarana yang sangat potensial untuk mengatasi keterbatasan informasi transportasi umum yang sebelumnya bersifat manual dan terfragmentasi. Aplikasi yang intuitif dan mudah digunakan akan meningkatkan adopsi masyarakat terhadap layanan digital[21].

Penelitian ini mengembangkan aplikasi *mobile* berbasis *Android* yang dirancang khusus untuk pengguna transportasi umum di Kota Tangerang. Pemilihan *platform mobile* memungkinkan penyampaian informasi yang cepat, akurat, dan dapat diakses kapan saja, sesuai dengan kebutuhan masyarakat urban yang dinamis.

2.2.4 Sistem Informasi dan Aksesibilitas Data

Sistem informasi adalah serangkaian komponen yang saling berinteraksi untuk mengumpulkan, mengolah, menyimpan, dan menyebarkan data agar dapat digunakan dalam pengambilan keputusan. Dalam praktiknya, sistem informasi berfungsi sebagai jembatan antara data mentah dan informasi yang dapat dimanfaatkan oleh pengguna[22]. Pada organisasi publik, sistem ini berfungsi untuk meningkatkan transparansi, efisiensi, dan layanan kepada masyarakat.

Aksesibilitas informasi merupakan faktor penting dalam sistem transportasi. Tanpa sistem informasi yang andal, data terkait jadwal dan rute akan sulit dipahami dan digunakan oleh masyarakat. Sistem informasi yang terintegrasi memungkinkan penyajian data transportasi secara terstruktur, baik melalui peta digital, tabel jadwal, maupun pemberitahuan perubahan layanan. Ini sangat membantu dalam menyesuaikan rencana perjalanan harian pengguna[23].

Dalam penelitian ini, sistem informasi berperan sebagai fondasi utama aplikasi yang dikembangkan. Informasi jadwal, rute, serta fitur pencarian dan penyimpanan favorit akan dikemas dalam sistem berbasis *mobile* yang mudah diakses oleh pengguna transportasi umum di Kota Tangerang.

2.2.5 Rancang Bangun Sistem dalam Penelitian Teknologi

Rancang bangun sistem adalah proses terstruktur dalam menciptakan sistem informasi, mulai dari identifikasi kebutuhan, perancangan fungsional, pengembangan antarmuka, hingga tahap implementasi dan pengujian. Pendekatan ini menekankan pada

dokumentasi, ketepatan analisis, dan desain yang sesuai dengan kebutuhan pengguna akhir[24]. Dalam konteks penelitian teknologi, rancang bangun menjadi metode penting untuk memastikan sistem dikembangkan berdasarkan landasan teoritis dan kebutuhan lapangan.

Dalam proses ini, aspek desain sistem meliputi perancangan struktur database, logika proses bisnis, alur pengguna, serta interaksi antarmuka. Tahapan ini juga mencakup simulasi dan uji coba awal agar sistem dapat diperbaiki sebelum diimplementasikan sepenuhnya. Metode pengembangan seperti *SDLC* atau *Agile* sering digunakan untuk membungkus proses rancang bangun secara terukur dan sistematis[25].

Penelitian ini menerapkan pendekatan rancang bangun berbasis *SDLC Waterfall* untuk memastikan setiap tahapan dari analisis hingga pengujian berjalan sesuai urutan dan terdokumentasi dengan baik. Hasil dari pendekatan ini adalah aplikasi mobile yang mampu menyajikan informasi penjadwalan transportasi umum secara lengkap dan *user-friendly*.

2.3 Teori tentang Framework/Algoritma yang digunakan

2.3.1 Pengertian Framework Pengembangan Sistem

Framework pengembangan sistem merupakan pedoman atau kerangka kerja konseptual yang digunakan dalam proses merancang, membangun, dan mengelola sistem informasi. *Framework* ini membantu dalam mengatur tahapan kerja agar lebih sistematis dan terukur. Selain itu, *framework* juga mempermudah dalam dokumentasi serta pengambilan keputusan selama siklus pengembangan berlangsung.

Secara umum, *framework* pengembangan sistem berisi metode, praktik terbaik, dan prosedur standar yang dapat diadopsi untuk mengurangi kesalahan, meningkatkan efisiensi tim, dan menghindari pengulangan pekerjaan. Salah satu manfaat utama dari *framework*

adalah kemampuannya untuk menyesuaikan tahapan pengembangan dengan ruang lingkup dan tujuan sistem yang akan dibangun[26].

Dalam penelitian ini, framework digunakan sebagai dasar dalam menyusun langkah-langkah pengembangan aplikasi, mulai dari analisis kebutuhan hingga pengujian. Hal ini bertujuan untuk memastikan aplikasi yang dihasilkan dapat dibangun secara sistematis, terdokumentasi dengan baik, serta sesuai dengan kebutuhan pengguna transportasi umum di Kota Tangerang.

2.3.2 System Development Life Cycle (SDLC)

System Development Life Cycle (SDLC) adalah suatu pendekatan terstruktur yang terdiri dari serangkaian tahapan sistematis untuk mengembangkan sistem perangkat lunak. SDLC memiliki berbagai model, di antaranya *Waterfall*, *Agile*, *Spiral*, dan *V-Model*. Masing-masing model memiliki keunggulan yang dapat disesuaikan dengan kebutuhan dan karakteristik proyek yang dijalankan [27].

SDLC digunakan sebagai standar dalam pengembangan sistem informasi untuk menghasilkan produk akhir yang berkualitas dan dapat diandalkan. Setiap fase dari SDLC saling terkait dan berfungsi untuk memastikan bahwa sistem dikembangkan secara efisien, mulai dari perencanaan, desain, implementasi, hingga pemeliharaan[28].

Dalam penelitian ini, pendekatan SDLC digunakan untuk membantu memetakan urutan kegiatan yang logis dan efisien. Dengan membagi proses pengembangan menjadi tahapan-tahapan yang jelas, SDLC membantu tim pengembang dalam menyusun prioritas pekerjaan dan menetapkan target waktu penyelesaian sistem.

2.3.3 SDLC Waterfall

Model *Waterfall* merupakan salah satu bentuk SDLC yang bersifat linier dan sekuensial, artinya setiap tahap harus diselesaikan sebelum tahap berikutnya dimulai. Tahapan tersebut meliputi: analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan.

Model ini cocok untuk proyek yang memiliki kebutuhan stabil dan ruang lingkup yang telah ditentukan sejak awal[29].

Keunggulan *Waterfall* terletak pada dokumentasi yang lengkap dan alur kerja yang jelas. Model ini memudahkan manajemen proyek dalam memantau kemajuan dan mengatur sumber daya. Namun, kelemahannya adalah tidak fleksibel terhadap perubahan mendadak di tengah siklus pengembangan[30].

Karena aplikasi transportasi umum dalam penelitian ini memiliki fitur dan kebutuhan yang sudah pasti sejak awal, pendekatan *Waterfall* dipilih agar setiap fase dapat dilaksanakan secara terstruktur dan terdokumentasi dengan baik. Ini memudahkan penyusunan laporan skripsi dan pengujian hasil akhir sistem.

2.3.4 Perbandingan SDLC *Waterfall* dan *Agile*

Metode *Agile* menawarkan pendekatan yang fleksibel, iteratif, dan kolaboratif dalam pengembangan perangkat lunak. *Agile* menekankan pada respons cepat terhadap perubahan dan komunikasi yang intensif antara pengembang dan pemilik produk. Pendekatan ini umum digunakan dalam proyek startup dan sistem yang kebutuhan fiturnya sering berubah[31].

Sebaliknya, model *Waterfall* menekankan pada perencanaan awal dan dokumentasi yang lengkap. *Waterfall* cocok untuk proyek bersifat prediktif, sedangkan *Agile* untuk proyek adaptif. Beberapa penelitian menemukan bahwa *Agile* unggul dalam kecepatan iterasi dan kepuasan pengguna, sedangkan *Waterfall* unggul dalam kontrol dokumentasi dan estimasi biaya yang lebih stabil [32].

Dalam konteks penelitian ini, *Agile* tidak digunakan karena sistem yang dikembangkan bersifat tetap dan tidak mengalami banyak perubahan fitur selama proses berlangsung. Dengan demikian, *Waterfall* dianggap lebih tepat karena mendukung proses skripsi yang membutuhkan dokumentasi lengkap dan tahapan yang sistematis.

Tabel 2. 2 Perbandingan Metode Agile dan Waterfall

Aspek	<i>Waterfall</i>	<i>Agile</i>
Pendekatan	Iteratif dan fleksibel	Linear dan sistematis
Dokumentasi	Minim, fokus pada komunikasi langsung	Lengkap dan terstruktur sejak awal
Perubahan Fitur	Mudah beradaptasi terhadap perubahan	Perubahan sulit dilakukan setelah tahap awal selesai
Kecocokan Proyek	Cocok untuk proyek dengan kebutuhan yang belum pasti atau sering berubah	Cocok untuk proyek dengan kebutuhan tetap dan terdefinisi sejak awal
Kelebihan	Cepat dalam iterasi, meningkatkan keterlibatan pengguna	Mudah dikontrol, estimasi biaya dan waktu lebih stabil
Kelemahan	Risiko tidak adanya dokumentasi yang cukup	Kurang fleksibel terhadap perubahan mendadak
Contoh Implementasi	Aplikasi startup, sistem dengan pengembangan berkelanjutan	Proyek akademik, sistem dengan ruang lingkup terbatas dan terencana

Pada tabel 2.2 memperlihatkan bahwa metode *Waterfall* lebih sesuai untuk konteks pengembangan aplikasi transportasi berbasis Android dalam penelitian ini. Hal ini karena proses skripsi membutuhkan dokumentasi yang sistematis, ruang lingkup proyek yang relatif tetap, dan kontrol yang baik atas tahapan implementasi. Meskipun *Agile* menawarkan fleksibilitas dan iterasi cepat, karakteristik proyek ini lebih condong ke arah prediktif, bukan adaptif.

2.3.5 Alasan Pemilihan SDLC *Waterfall* dalam Penelitian ini

Pemilihan SDLC *Waterfall* pada penelitian ini dilatarbelakangi oleh perlunya proses pengembangan yang terstruktur dan terdokumentasi. Lingkup sistem telah ditentukan sejak awal dan tidak mengalami perubahan signifikan selama tahap pengembangan. Hal ini selaras dengan prinsip kerja *Waterfall* yang mendukung pengembangan secara bertahap dan berurutan[33].

Waterfall juga memungkinkan dokumentasi lengkap pada setiap tahap, mulai dari analisis kebutuhan, perancangan antarmuka pengguna, hingga proses pengujian sistem. Dokumentasi ini sangat penting dalam penyusunan skripsi sebagai bukti proses ilmiah dan validasi terhadap hasil yang diperoleh. Tahapan yang rapi juga membantu peneliti dalam menjelaskan kontribusi teknis secara sistematis.

Dengan demikian, SDLC *Waterfall* memberikan kontrol yang baik dalam setiap fase pengembangan aplikasi, serta mendukung kebutuhan akademis penulisan skripsi. Selain itu, model ini memberikan kejelasan dalam manajemen waktu dan pembagian pekerjaan selama proyek berlangsung.

2.4 Teori tentang tools/software yang digunakan

2.4.1 *Android Studio*

Android Studio adalah lingkungan pengembangan terintegrasi (IDE) resmi yang disediakan oleh *Google* untuk pengembangan aplikasi Android. IDE ini dibangun di atas platform *IntelliJ IDEA* dan mendukung berbagai fitur seperti *code editor* pintar, *visual layout editor*, emulator *Android*, serta integrasi dengan sistem version control seperti *Git*. *Android Studio* memudahkan pengembang untuk membangun, menguji, dan mendistribusikan aplikasi secara efisien[34].

Dengan antarmuka yang *user-friendly* dan fitur *debugging* yang mendalam, *Android Studio* menjadi standar industri dalam

pengembangan aplikasi berbasis *Android*. Selain itu, *Android Studio* mendukung integrasi dengan berbagai library eksternal dan API yang digunakan dalam pengembangan modern. Dukungan terhadap berbagai versi *Android* juga menjadi kelebihan tersendiri [35].

Dalam penelitian ini, *Android Studio* digunakan untuk membangun aplikasi *mobile* transportasi umum yang berjalan di sistem operasi *Android*. IDE ini membantu peneliti untuk menyusun struktur proyek, menulis kode dalam *Java*, serta menjalankan pengujian aplikasi di emulator maupun perangkat fisik.

2.4.2 Java

Java adalah bahasa pemrograman berorientasi objek (OOP) yang bersifat *platform-independent* karena menggunakan *Java Virtual Machine* (JVM). *Java* banyak digunakan untuk pengembangan aplikasi *mobile* *Android* karena kestabilannya, skalabilitasnya, dan komunitas pengembangnya yang besar. *Java* menyediakan berbagai *library* standar untuk menangani *input/output*, *GUI*, *database*, dan jaringan[36].

Java memiliki sintaks yang relatif mudah dipahami dan struktur kode yang jelas. Fitur seperti *garbage collection* dan *exception handling* menjadikan *Java* cocok untuk membangun sistem yang stabil dan mudah dipelihara. Oleh karena itu, banyak pengembang *Android* masih memilih *Java* sebagai bahasa utama meskipun *Kotlin* juga mulai banyak digunakan [37].

Penelitian ini menggunakan *Java* sebagai bahasa pemrograman utama dalam membangun logika aplikasi penjadwalan transportasi. Dengan *Java*, peneliti dapat mengimplementasikan antarmuka pengguna, mengatur koneksi API, serta mengelola struktur data internal yang dibutuhkan untuk menampilkan informasi jadwal secara dinamis.

Untuk memperjelas kontribusi *Java* dalam pengembangan sistem ini, disusun Tabel 2.4 berikut. Tabel ini menguraikan kelebihan utama

Java yang membuatnya relevan digunakan dalam aplikasi *mobile Android*.

Tabel 2. 3 Tabel Kelebihan Java

No	Kelebihan Java	Penjelasan Singkat
1	Portabilitas Tinggi	Java dapat dijalankan di berbagai platform menggunakan JVM
2	Struktur Berorientasi Objek	Mendukung modularitas dan reusability dalam pengembangan aplikasi
3	Komunitas Besar dan Dokumentasi Luas	Banyak tutorial dan pustaka tersedia untuk pengembangan Android
4	Stabilitas dalam Proyek Skala Menengah	Cocok untuk proyek akademik dan produksi karena performa yang andal

Dari tabel di atas dapat disimpulkan bahwa Java memiliki kelebihan yang mendukung produktivitas dan efisiensi dalam pengembangan aplikasi Android, terutama dari sisi skalabilitas, dokumentasi, dan struktur yang terorganisir.

2.4.3 Visual Studio Code

Visual Studio Code (VS Code) adalah code editor ringan namun kuat yang dikembangkan oleh *Microsoft*. Meskipun bukan IDE penuh seperti *Android Studio*, VS Code sangat fleksibel karena mendukung banyak bahasa pemrograman dan memiliki ekosistem ekstensi yang luas. Fitur seperti *IntelliSense*, *Git integration*, dan *debugging* membuatnya menjadi pilihan populer di kalangan developer[38].

Dalam proyek yang melibatkan pengembangan *web* atau *backend* (seperti *RESTful API*), VS Code sering digunakan karena ringan dan

cepat. Editor ini juga dapat dikustomisasi dengan berbagai plugin untuk PHP, JavaScript, dan framework lain seperti Laravel, Node.js, atau Flask. VS Code berjalan di semua *platform* utama (Windows, macOS, Linux) [39].

Dalam konteks penelitian ini, VS Code digunakan untuk mengembangkan bagian backend dari aplikasi yang mengatur penyimpanan data, koneksi ke database, dan penyediaan API untuk konsumsi oleh aplikasi Android. Integrasi VS Code dengan MySQL dan Postman juga membantu dalam pengujian sistem.

2.4.4 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah alat visual yang digunakan untuk merancang struktur basis data dan hubungan antar entitas. ERD sangat penting dalam tahap perancangan sistem karena membantu pengembang memahami dan memodelkan kebutuhan data secara logis sebelum membangunnya di level fisik[40].

ERD terdiri dari entitas (tabel), atribut (kolom), dan relasi (hubungan antar entitas). Model ini memastikan struktur data yang efisien, menghindari redundansi, dan mendukung normalisasi. Dalam pengembangan aplikasi berskala kecil hingga besar, penggunaan ERD membantu mempercepat proses desain basis data yang akurat dan mudah dipelihara[41].

Penelitian ini menggunakan ERD untuk memetakan struktur database transportasi, termasuk entitas pengguna, rute, jadwal, dan data favorit. ERD dibuat sebelum implementasi MySQL agar skema database dapat langsung diterapkan sesuai rancangan awal. Berikut ini adalah entitas utama yang digunakan dalam sistem:

Tabel 2. 4 Daftar Entitas dan Deskripsi dalam ERD

No	Nama Entitas	Atribut Utama	Deskripsi Singkat
----	--------------	---------------	-------------------

1	Pengguna	ID_Pengguna, Nama, Email	Menyimpan informasi akun pengguna
2	Transportasi	ID_Transportasi, Nama, Jenis	Data kendaraan umum seperti bus atau MRT
3	Rute	ID_Rute, TitikAwal, TitikTujuan	Informasi rute perjalanan tiap transportasi
4	Jadwal	ID_Jadwal, Hari, Jam, ID_Rute	Data jadwal transportasi per rute
5	Favorit	ID_Favorit, ID_Pengguna, ID_Jadwal	Data relasi antara pengguna dan jadwal

Tabel ini menunjukkan struktur logis dari sistem basis data yang dikembangkan, memastikan bahwa seluruh data pengguna dan operasional transportasi umum dapat dikelola secara terstruktur dan saling terhubung.

2.4.5 *Unified Modeling Language*

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan dalam perancangan sistem perangkat lunak. UML membantu menggambarkan struktur dan perilaku sistem melalui berbagai jenis diagram seperti *use case*, *class*, *activity*, dan *sequence diagram*[42].

UML memungkinkan pengembang untuk mendokumentasikan sistem secara sistematis, mempermudah komunikasi antar tim, serta mengidentifikasi kebutuhan fungsional dan non-fungsional sistem. Pemodelan dengan UML sangat penting dalam proyek akademik

maupun industri untuk mendefinisikan skenario pengguna dan alur sistem dengan jelas[43].

Dalam penelitian ini, UML digunakan untuk menggambarkan berbagai aspek sistem transportasi umum yang dikembangkan, seperti alur login pengguna, pemilihan rute, dan penyimpanan jadwal favorit. Berikut adalah jenis diagram UML yang digunakan dalam sistem:

Tabel 2. 5 Jenis Diagram UML

No	Jenis Diagram	Fungsi dalam Sistem	Keterangan Tambahan
1	<i>Use Case Diagram</i>	Menjelaskan interaksi pengguna dengan sistem	Termasuk fitur login & pencarian
2	<i>Activity Diagram</i>	Menggambarkan alur logika dari suatu proses	Seperti pemesanan atau pencarian
3	<i>Class Diagram</i>	Menunjukkan struktur data dan relasi antar kelas	Digunakan untuk pemodelan objek data

Dengan penggunaan diagram-diagram tersebut, sistem dapat divisualisasikan secara menyeluruh sebelum proses implementasi teknis, sehingga mendukung proses analisis kebutuhan dan perancangan sistem secara lebih sistematis.

2.4.6 XAMPP

Xampp adalah paket perangkat lunak bebas yang menyediakan lingkungan *server* lokal lengkap untuk pengembangan web. XAMPP terdiri dari Apache (*web server*), MySQL/MariaDB (basis data), PHP, dan Perl, yang dikemas menjadi satu dan dapat dijalankan di sistem operasi Windows, macOS, dan Linux. Alat ini sangat populer di kalangan

pengembang karena instalasinya yang mudah dan antarmuka kontrolnya yang sederhana[44].

XAMPP digunakan secara luas untuk mengembangkan dan menguji aplikasi web secara lokal sebelum diunggah ke server produksi. Dengan adanya XAMPP, pengembang dapat menjalankan server lokal tanpa harus mengonfigurasi komponen secara manual. MySQL atau MariaDB yang terintegrasi di dalamnya memungkinkan pengelolaan database secara langsung melalui phpMyAdmin[45].

Dalam penelitian ini, XAMPP digunakan untuk menyimpan dan mengelola database sistem informasi transportasi yang diakses oleh aplikasi Android melalui API. Backend dan layanan API yang dibangun menggunakan PHP dapat dijalankan secara lokal menggunakan server Apache dari XAMPP. Selain itu, phpMyAdmin sangat membantu dalam proses pembuatan tabel, query, dan pemantauan data secara langsung.

2.4.7 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side* yang dirancang khusus untuk pengembangan web. PHP mampu berjalan di berbagai platform dan mudah diintegrasikan dengan berbagai sistem manajemen basis data seperti MySQL dan MariaDB. Karena bersifat *open source* dan memiliki dokumentasi yang luas, PHP menjadi salah satu bahasa backend yang paling banyak digunakan di dunia[46].

PHP cocok untuk membangun RESTful API, *login* autentikasi, pengelolaan data, dan komunikasi antara *server* dan *client*. Dalam lingkungan lokal seperti XAMPP, PHP bekerja secara optimal karena terintegrasi langsung dengan Apache dan MySQL, sehingga mempermudah proses *debugging* dan implementasi backend tanpa harus menggunakan server produksi[47].

Dalam penelitian ini, PHP digunakan untuk membangun API yang menghubungkan aplikasi Android dengan basis data transportasi. Script PHP menangani request seperti pencarian rute, penyimpanan data

favorit, dan pemanggilan jadwal dari server lokal. Integrasi antara PHP, MySQL, dan *Android* menjadikan alur kerja sistem menjadi responsif dan efisien.

Untuk menggambarkan arsitektur sistem secara visual, digunakan beberapa jenis diagram UML. Tabel berikut merinci tipe-tipe diagram UML yang digunakan dalam penelitian serta fungsinya masing-masing.

Tabel 2. 6 Fungsi dan Kelebihan PHP

No	Fitur PHP	Manfaat dalam Penelitian
1	Kompatibilitas Tinggi	Mudah dijalankan di server lokal seperti XAMPP
2	Terintegrasi dengan MySQL	Mudah mengelola basis data jadwal dan pengguna
3	Efisiensi untuk RESTful API	Menyediakan endpoint untuk komunikasi dengan aplikasi Android
4	Sintaks yang Ringan dan Familiar	Mempercepat proses pengembangan backend dan pengujian lokal

Melalui tabel ini, jelas bahwa PHP menyediakan infrastruktur backend yang ringan, fleksibel, dan sangat terintegrasi dengan tools lain yang digunakan dalam sistem, menjadikannya ideal untuk sistem terdistribusi berskala kecil hingga menengah.

2.4.8 Figma

Figma adalah perangkat lunak desain antarmuka berbasis cloud yang memungkinkan kolaborasi real-time antara *desainer* dan *developer*. Figma memungkinkan pembuatan *wireframe*, prototipe, serta desain antarmuka pengguna (UI) secara interaktif dan terstruktur. Karena berbasis web, Figma dapat diakses dari berbagai *platform* tanpa perlu instalasi software tambahan[48].

Dalam konteks pengembangan aplikasi *mobile*, Figma sangat membantu dalam merancang tampilan layar aplikasi sebelum tahap implementasi kode dimulai. Fitur seperti komponen, *auto layout*, dan *prototyping* membuat desainer dapat membuat desain yang konsisten dan dapat diuji secara interaktif oleh stakeholder maupun pengguna[49].

Dalam penelitian ini, Figma digunakan untuk mendesain antarmuka aplikasi transportasi umum berbasis Android. Proses desain mencakup halaman login, beranda, hasil pencarian rute, hingga detail jadwal transportasi. Penggunaan Figma bertujuan untuk memvisualisasikan pengalaman pengguna sebelum implementasi teknis dilakukan.

Dalam proses pengembangan aplikasi transportasi umum berbasis *Android*, berbagai tools dan perangkat lunak digunakan untuk menunjang setiap tahapan mulai dari desain antarmuka, pengembangan kode, hingga pengujian sistem dan pengelolaan basis data. Penggunaan tools yang tepat akan memengaruhi efisiensi, akurasi, dan kemudahan dalam proses pengembangan.

Tabel 2.7 berikut merangkum seluruh perangkat lunak dan *framework* yang digunakan dalam penelitian ini, termasuk fungsi utama serta keterlibatannya dalam proyek pengembangan aplikasi.

Tabel 2. 7 Rangkuman Tools/software yang Digunakan

No	Tools/Software	Fungsi Utama	Keterlibatan dalam Penelitian
1	<i>Android Studio</i>	IDE untuk pengembangan aplikasi <i>Android</i>	Membangun aplikasi <i>mobile</i>
2	<i>Java</i>	Bahasa pemrograman untuk <i>Android</i>	Logika dan antarmuka aplikasi
3	<i>Visual Studio Code</i>	Editor kode untuk pengembangan web/backend	<i>Backend</i> dan API dengan PHP

4	<i>PHP</i>	Bahasa <i>backend web</i>	Menyusun API & koneksi <i>database</i>
5	<i>ERD</i>	Pemodelan data dan hubungan antar entitas	Desain database transportasi
6	<i>UML</i>	Visualisasi struktur dan alur sistem	Use case, activity, class diagram
7	<i>XAMPP</i>	<i>Server</i> lokal (Apache, MySQL, PHP)	Menyimpan & mengakses data sistem secara lokal
8	<i>Figma</i>	desain UI Tools berbasis web/cloud	Mendesain antarmuka pengguna aplikasi

Penggunaan tools/software di atas disesuaikan dengan kebutuhan teknis dan alur pengembangan sistem berbasis *SDLC Waterfall*. Setiap tools berkontribusi secara spesifik dalam menunjang tahapan analisis, desain, implementasi, dan pengujian aplikasi. Dengan pemilihan alat bantu yang tepat, pengembangan aplikasi dapat berjalan lebih terstruktur dan terdokumentasi dengan baik sesuai standar ilmiah dalam penelitian ini.

