

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian yang mendukung Aplikasi Jasa Laundry Dengan Fitur Pemesanan Dan Notifikasi Status Laundry Berbasis Android disajikan pada :

Tabel 2. 1 Penelitian Terdahulu

No.	Penulis	Judul	Hasil Penelitian
1.	(Santoso, 2020)	Sistem Informasi Geografis Pemetaan Mitra Pengguna Aplikasi Laundry di PT Tenten Digital Indonesia Berbasis Web dan Mobile	PT Tenten Digital Indonesia mengembangkan sistem web dan mobile untuk pemetaan mitra laundry menggunakan SDLC, UML, dan Google Maps API. Sistem diuji dengan black box dan terbukti bebas bug.
2.	(Sari & Setiawan, 2022)	Aplikasi D'laundry Berbasis Android Menggunakan Model Design Thinking	Aplikasi rental mobil berbasis REST API memudahkan booking tanpa harus datang ke lokasi, dengan pencatatan otomatis yang mengurangi kesalahan dan mempercepat pelayanan.
3.	(Antoni et al., 2022)	Aplikasi Laundry Berbasis Website Dan Android Pada T & F laundry	sistem berbasis web untuk operasional karyawan dan aplikasi Android untuk layanan antar-jemput, memudahkan konsumen dan meningkatkan efisiensi.
4.	(Rafi Taqiyuddin & Indryanti, 2024)	Perancangan Aplikasi Laundry Berbasis Android Dengan Menggunakan Metode Design Thinking	Penelitian ini merancang aplikasi laundry dengan metode Design Thinking berdasarkan survei pengguna. Evaluasi menghasilkan skor <i>SUS 84,17 (Grade B, Excellent)</i> , membuktikan aplikasi mudah digunakan dan diterima dengan baik.
5.	(Persada, 2020)	Pembangunan Aplikasi Mobile Pencarian Dan Pemesanan Jasa Laundry Berbasis Lokasi	Aplikasi Laundree dan LaundreePOS berbasis <i>Flutter-Firestore</i> efektif untuk pemesanan dan pengelolaan laundry, namun masih perlu fitur rating dan notifikasi.

2.2. Laundry

Menurut penelitian dari (Hasanah et al., 2021), laundry adalah layanan mencuci pakaian dan kain menggunakan air, deterjen, pelembut dan pewangi pakaian. Layanan laundry bukan hanya tempat untuk mencuci pakaian, tetapi juga tempat untuk merawat pakaian agar lebih bersih dan tahan lama.

2.3. Aplikasi

Menurut penelitian dari (Azis et al., 2020), Aplikasi berasal dari kata *application* yang artinya penerapan, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju.

2.4. Android

Menurut penelitian dari (Azis et al., 2020), Android adalah sebuah sistem operasi perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

2.5. Visual Studio Code

Menurut penelitian dari (Sofi & Dharmawan, 2022), Visual Studio Code adalah editor kode sumber yang ringan namun kuat yang berjalan di desktop dan tersedia untuk Windows, macOS, dan Linux. Muncul dengan dukungan built-in untuk JavaScript, TypeScript dan Node.js dan memiliki ekosistem ekstensi yang kaya untuk bahasa lain (seperti C ++, C #, Java, Python, PHP, Go) dan runtime (seperti .NET dan Unity).

2.6. BlackBox

Menurut (Jibril et al., 2024), *black box testing* adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas aplikasi tanpa mengetahui struktur internal kode. Tujuannya untuk memastikan bahwa fitur sistem berfungsi sesuai dengan spesifikasi.

Dalam penelitian ini, metode black box testing digunakan untuk menguji fungsionalitas utama aplikasi WashClean, seperti fitur pemesanan layanan laundry, notifikasi berdasarkan status, dan sinkronisasi data melalui REST API. Pengujian dilakukan dari sisi pengguna dan mitra untuk memastikan aplikasi bekerja sesuai kebutuhan pengguna akhir.

2.7. Flutter

Menurut penelitian dari (Bayu et al., 2023), Flutter menggunakan bahasa pemrograman Dart untuk membuat aplikasi. Pemrograman Dart memiliki beberapa fitur yang sama dengan bahasa pemrograman lain, seperti Kotlin dan Swift, dan dapat dicompile menjadi kode JavaScript. Flutter terutama dioptimalkan untuk aplikasi seluler 2D yang dapat berjalan di platform Android dan iOS. Kita juga dapat menggunakannya untuk membangun aplikasi berfitur lengkap, termasuk kamera, penyimpanan, geolokasi, jaringan, SDK pihak ketiga, dan banyak lagi.

2.8. Dart

Menurut (Bayu et al., 2023), dart adalah bahasa pemrograman open source, dan berorientasi objek dengan sintaks gaya C yang dikembangkan oleh Google pada tahun 2011. Tujuan pemrograman Dart adalah untuk membuat antarmuka pengguna frontend untuk web dan aplikasi seluler. Dart sedang dalam pengembangan aktif, dikompilasi ke kode mesin asli untuk membangun aplikasi seluler, terinspirasi oleh

bahasa pemrograman lain seperti Java, JavaScript, C#. Karena Dart adalah bahasa yang dikompilasi sehingga Anda tidak dapat mengeksekusi kode Anda secara langsung. Sebaliknya, kompilator mem-parsingnya dan mentransfernya ke dalam kode mesin.

2.9. Waterfall

Menurut (Rahman, 2021), Requirement Analysis (Analisis Kebutuhan): Mengidentifikasi kebutuhan perangkat lunak dengan berkomunikasi langsung dengan pengguna melalui wawancara, survei, atau diskusi.

1. *Design System* (Desain Sistem): Spesifikasi kebutuhan sistem untuk menentukan perangkat keras dan perangkat lunak secara menyeluruh.
2. *Coding* (Pengkodean): Pengembangan unit kecil perangkat lunak (program) yang nantinya akan diintegrasikan.
3. *Testing* (Pengujian): Seluruh unit diintegrasikan dan diuji untuk memastikan tidak ada kegagalan atau kesalahan pada sistem.
4. *Maintenance* (Pemeliharaan): Memperbaiki kesalahan yang ditemukan setelah sistem berjalan dan menambahkan fitur baru sesuai kebutuhan pengguna.

2.10. Figma

Menurut (Muhyidin et al., 2020), Figma adalah alat desain antarmuka (UI) berbasis cloud yang memungkinkan kolaborasi secara real-time antar desainer. Figma digunakan untuk membuat prototipe aplikasi, mockup tampilan web atau mobile, serta mendesain user interface yang interaktif.

Dalam rancang bangun WashClean, Figma digunakan pada tahap desain antarmuka pengguna, khususnya untuk merancang UI aplikasi mobile dan

dashboard web admin. Figma dipilih karena kemampuannya mendukung desain kolaboratif dan prototyping yang cepat, sehingga memudahkan dalam proses validasi desain terhadap kebutuhan pengguna.

2.11 Data Flow Diagram (DFD)

Menurut (Paillin & Widiatmoko, 2021), Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. DFD menggambarkan penyimpanan data dan proses yang mentransformasikan data.