

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Teori Terkait**

Timothy Priambodo Hartono dan rekan-rekannya melakukan penelitian yang mencakup desain dan implementasi sistem informasi untuk mengolah data transaksi melalui sebuah situs web untuk usaha laundry di Ester. Mereka menggunakan berbagai teknologi termasuk *Codeigniter*, *MySQL*, dan, *Javascript*. Metode yang digunakan dalam penelitian ini adalah metode Waterfall [2].

Penelitian yang dilakukan oleh Nanang Hoesen dkk membahas pengembangan aplikasi layanan jasa laundry berbasis android dengan tujuan meningkatkan kualitas layanan di industri laundry pos. aplikasi ini dirancang untuk menyediakan berbagai fitur termasuk sistem *Point Of Sales*, pelacakan paket, dan layanan antar jemput laundry melalui aplikasi *android* [5].

Bohati Mulyadi dan timnya melakukan penelitian terkait dengan pengembangan aplikasi sistem pemesanan jasa laundry (*e-laundry*) berbasis Android yang dibangun dengan bahasa pemrograman *JAVA* dan menggunakan *MySQL* sebagai basis datanya. Aplikasi ini dilengkapi dengan berbagai fitur, termasuk penggunaan *GPS* untuk navigasi yang didukung oleh lebih dari 24 satelit *MEO (Medium Earth Orbit)*, serta layanan berbasis lokasi yang memungkinkan pengguna untuk menemukan tempat-tempat tertentu seperti SPBU dan ATM terdekat [4].

Muhammad Yasin Simargolang dan rekan-rekannya telah melakukan penelitian mengenai pengembangan aplikasi layanan jasa laundry berbasis web yang diimplementasikan di Pelangi Laundry Kisaran. Aplikasi ini dibangun menggunakan bahasa pemrograman *PHP* dan *MySQL* digunakan sebagai sistem basis datanya. Sistem ini dilengkapi dengan fitur pesan antar serta pelacakan paket untuk meningkatkan kualitas layanan pada Pelangi Laundry Kisaran [6].

Penelitian yang dilakukan oleh Nur Aini dkk membahas pembangunan sistem informasi strategis *CRM* dengan menggunakan *framework Codeigniter*, yang diimplementasikan di Laundry Bang Tris. Penelitian ini mengadopsi metode kualitatif dan menerapkan strategi *CRM (Customer Relationship Management)*, yang menghasilkan sebuah situs web sistem informasi laundry yang dibangun menggunakan *Framework Codeigniter* dan *MySQL* [7].

Pada penelitian yang dilakukan oleh Tarisno Amijoyo dkk (2023), yang memfokuskan pada Analisa Pemilihan Aplikasi Laundry Digital yang dilakukan menggunakan metode *AHP (Analytic Hierarchy Process)*, memiliki tujuan untuk mempermudah dalam mengambil keputusan memilih suatu aplikasi laundry digital yang sesuai dan efisien. Penelitian ini dilakukan untuk melakukan perbandingan bobot prioritas dari setiap kriteria yang telah ditentukan, dan melakukan analisis hasil keputusan pemilihan aplikasi menggunakan *AHP* [8].

Pada penelitian yang dilakukan oleh Hadi Gunawan dkk (2024), melakukan penelitian serta melakukan rancang bangun untuk sebuah aplikasi laundry berbasis website, yang bertujuan untuk memudahkan dalam melakukan rekapitulasi dan proses rekam transaksi yang berfokus pada Oasis Laundry.

Penelitian ini menggunakan metode waterfall dan menggunakan beberapa teknologi dalam pengembangan aplikasi web laundry digital, diantaranya menggunakan bahasa pemrograman *PHP* dengan menggunakan *Framework Laravel* serta menggunakan *MySQL* dan *XAMPP* untuk basis data dan *web servernya* [9].

Penelitian yang dilakukan oleh Rene Matthew Winarto dkk (2023), melakukan penelitian dan perancangan suatu aplikasi laundry berbasis android yang dibuat untuk Nalendra Laundry. Dalam perancangannya, aplikasi ini menggunakan struktur dari *HTML5* dan menggunakan *CSS* yang dibantu dengan *Framework CSS* yaitu *Bootstrap* yang bertugas untuk mempercantik tampilan dari aplikasi. Aplikasi ini bertujuan untuk memudahkan Nalendra Laundry dalam melakukan transaksi yang biasa dilakukan secara manual sehingga bisa dilakukan secara efisien [10].

Penelitian yang dilakukan oleh Adilla Syafira Putri dkk (2023), yang melakukan penelitian serta perancangan aplikasi *E-Commerce* berbasis Android pada toko Suryamart. Aplikasi ini dirancang menggunakan metode waterfall untuk meminimalisir kesalahan dan mempercepat proses pengembangan. Aplikasi ini bertujuan untuk membantu admin Suryamart dalam melakukan transaksi secara *online*. Untuk menunjangnya, aplikasi ini dibangun menggunakan bahasa pemrograman Dart dengan menggunakan *framework Flutter* [11].

Dalam referensi di atas, sudah banyak penelitian yang dilakukan seputar aplikasi laundry, baik yang dirancang berbasis website maupun berbasis android. Namun, dalam penelitian tugas akhir ini dengan judul Aplikasi Laundry Digital Berbasis Mobile Menggunakan *Flutter* ini dilakukan untuk mengembangkan serta menyempurnakan aplikasi dari penelitian sebelumnya.

## **2.2. Landasan Teori**

### **2.2.1. Pengertian Aplikasi**

Aplikasi merupakan perangkat lunak yang digunakan untuk memenuhi berbagai kebutuhan manusia dalam kegiatan sehari-hari. Aplikasi ini dapat diakses dan digunakan melalui berbagai perangkat seperti *smartphone*, *tablet*, dan komputer sesuai dengan *preferensi* pengguna.

Aplikasi memiliki banyak fungsi seperti pengolah kata, pengolah gambar, pengeditan video, *game*, *social media*, dan lain-lain. Aplikasi sendiri dibuat sedemikian rupa sesuai dengan fungsi dan kebutuhan pengguna, agar bisa mempermudah dalam melakukan suatu tugas tertentu.

Aplikasi merupakan program-program yang dibuat oleh suatu perusahaan yang beroperasi diberbagai bidang seperti pertokoan, industri kecil, industri berat hingga industri yang besar. [12]

### **2.2.2. Dart**

*Dart* adalah sebuah bahasa pemrograman yang dikembangkan oleh *Google*, yang dirancang oleh Lars Bak dan Kasper Lund. Bahasa pemrograman ini mengikuti konsep Pemrograman Berbasis Objek, dengan model penulisan yang menyerupai bahasa pemrograman C. [12]

*Dart* populer dalam 2 tahun terakhir ini, karena sering digunakan dalam pengembangan aplikasi mobile yang bisa dilakukan secara *Cross*

*Platform*, sehingga memungkinkan seorang *developer* melakukan pengembangan aplikasi pada *platform* yang berbeda namun dengan basis kode yang sama.

### 2.2.3. MySQL

*Structured Query Language* yang umumnya dikenal sebagai *MySQL*. Merupakan *Server Database* yang bersifat *Open Source* dan biasa digunakan untuk berbagai aplikasi. *MySQL* sendiri adalah sebuah implementasi dari sistem *RDBMS (Relational Database Management System)* yang di publikasi oleh *GPL (General Public License)*. *MySQL* ini mendukung operasi transaksi *database* yang handal dan cepat. [13]

### 2.2.4. Visual Studio Code

*Visual Studio Code (VS Code)* adalah sebuah editor teks yang cukup populer digunakan oleh *developer* dalam mengembangkan aplikasi. *Visual Studio Code* ini memiliki banyak fitur yang bisa membantu *developer* dalam mengembangkan aplikasi lebih cepat namun dengan beban yang relatif ringan.

### 2.2.5. Node JS

*Node JS* adalah salah satu *runtime Javascript* yang cukup populer digunakan untuk membuat sebuah aplikasi atau sebuah *REST API (Application Programming Interface)*, [14]

*Node JS* ini menawarkan fitur unggulan berupa *runtime* tanpa browser, sehingga kita bisa mengakses kode-kode *Javascript* dimana saja tanpa menggunakan browser. *Node JS* ini menggunakan arsitektur *V8 Engine* yang membuat kode *Javascript* bisa digunakan di mana saja.

### **2.2.6. Flutter**

*Flutter* adalah sebuah *Framework* atau kerangka kerja yang dikembangkan oleh *Google* untuk mengembangkan aplikasi mobile dari sisi *Front End*. *Flutter* sendiri berjalan menggunakan bahasa pemrograman *Dart*. *Flutter* memiliki keunggulan dalam melakukan pengembangan aplikasi mobile karena *Flutter* bisa melakukan pengembangan aplikasi mobile secara *Cross Platform* sehingga *developer* hanya perlu menuliskan satu basis kode untuk dua *platform* yang berbeda. [15]

### **2.2.7. Database**

*Database* atau basis data merupakan suatu sekelompok data yang tersimpan dalam suatu sistem dengan cara tertentu, di mana data tersebut memiliki hubungan antara satu dengan yang lainnya, dengan tujuan untuk mempermudah pengelolaan data. [13]

### **2.2.8. Teori Pengujian *Black Box***

Metode pengujian *Black Box* adalah metode pengujian yang biasa dipakai dalam menguji sebuah detail *software*. Pengujian ini hanya memeriksa nilai *output* dari sebuah *software* tanpa upaya untuk mengetahui kode program apa yang dipakai dalam *software*.

### 2.2.9. UML (Unified Modeling Language)

*Unified Modelling Language (UML)* adalah sebuah bahasa grafis yang digunakan untuk merancang, mendokumentasikan, dan memahami sistem perangkat lunak. Dengan menggunakan notasi yang konsisten, *UML* memungkinkan pengembangan perangkat lunak bisa digambarkan secara terstruktur. [16]

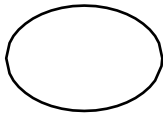

Terdapat empat model UML yang paling sering digunakan untuk menggambarkan suatu desain sistem *Usecase diagram*, *Class diagram*, *Behavioral State machine diagram*, dan juga *Sequence diagram*. Teknik-teknik pemodelan *Unified Modeling Language* ini disebut juga dengan 4 teknik dasar. Dalam proyek berorientasi objek, keempat teknik UML ini sangat mendominasi penggunaannya.


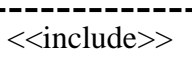

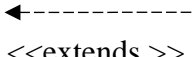
Dalam perancangan sistem terdapat *UML* yang sering digunakan sebagai berikut:

#### 1. *UseCase Diagram*

Dalam pengembangan sistem informasi, *UseCase Diagram* digunakan untuk memodelkan perilaku atau fungsi dari sistem informasi. Diagram ini juga memiliki karakteristik yang statis. Untuk simbol *UseCase Diagram* bisa di lihat pada Tabel 2.1.

Tabel 2. 1 Simbol *UseCase Diagram*.



No	Simbol	Keterangan
1.		<i>Use Case</i> merupakan bentuk untuk menggambarkan interaksi antara sistem dengan <i>Actor</i> .
2.		<i>Actor</i> merupakan himpunan peran untuk berinteraksi dengan <i>UseCase</i> .

3.		<i>Association</i> merupakan garis yang menggambarkan hubungan antara satu objek dengan objek lainnya.
4.		<i>Include</i> , merupakan gambaran jika <i>usecase</i> dipanggil oleh <i>usecase</i> lain.
5.		<i>Dependency</i> merupakan garis panah yang menunjukkan jika aktor berinteraksi secara pasif.
6.		<i>Extend</i> , merupakan gambaran jika memperluas <i>usecase</i> target.

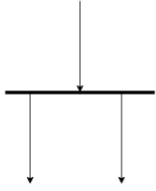
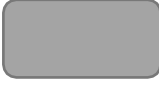
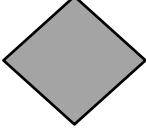
## 2. Activity Diagram

Diagram Aktivitas, juga dikenal sebagai *Activity Diagram*, adalah suatu representasi statis yang mengilustrasikan aktivitas-aktivitas dalam sebuah sistem bisnis. Untuk simbol dari *Activity Diagram* bisa dilihat pada Tabel 2.2.

Tabel 2. 2 Simbol *Activity Diagram*.

No	Simbol	Keterangan
1.		<i>End Point</i> atau <i>Final Node</i> adalah representasi dari hasil akhir dari suatu aktivitas.
2.		<i>Start Point</i> adalah awal dari suatu aktivitas yang ditempatkan di pojok kiri atas.


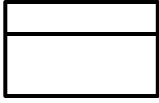
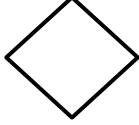


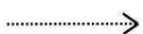



3.		<i>Fork</i> atau <i>Join</i> , digunakan untuk memarallelkan suatu kegiatan atau penggabungan 2 kegiatan paralel menjadisatu.
4.		<i>Activity</i> , merupakan gambaran dari suatu proses.
5.		<i>Decision</i> , merupakan pilihan pengambilan suatu keputusan <i>false or true</i> .

### 3. *Class Diagram*

Diagram Kelas atau biasa dikenal sebagai *Class Diagram* adalah representasi statis yang menampilkan kumpulan kelas, antarmuka, dan hubungan di antara mereka. Berikut simbol dari *Class Diagram* yang dapat dilihat pada Tabel 2.3.

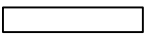
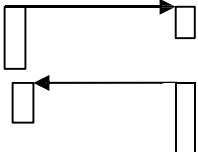





Tabel 2. 3 Simbol *Class Diagram*.

No	Gambar	Keterangan
1.		<i>Generallization</i> adalah ketika objek turunan mengadopsi perilaku dan struktur data dari objek induknya.
2.		<i>Class</i> , adalah kumpulan objek yang saling berbagi.
3.		<i>Nary Association</i> , digunakan untuk asosiasi terhindar dengan objek lainnya.
4.		<i>Collaboration</i> adalah representasi dari urutan aksi yang dijelaskan oleh suatu sistem yang memiliki dampak yang dapat diukur bagi aktor yang terlibat.
5.		Ini merupakan operasi yang dapat dilakukan secara sah oleh suatu objek.
6.		Ini merupakan garis panah yang menunjukkan interaksi pasif para aktor.
7.		<i>Association</i> merupakan garis yang menghubungkan objek satu dengan objek lainnya.

#### 4. Sequence Diagram

*Sequence Diagram* adalah suatu diagram yang dibuat untuk menekankan interaksi urutan waktu dan pesan pada suatu sistem. *Sequence diagram* ini berinteraksi pada sumbu X dan akan menempatkan pesan antar objek pada sumbu Y, dan waktu dalam diagram ini digambarkan dari atas ke bawah. Komponen dari *Sequence diagram* ini dapat dilihat pada Tabel 2.5.

Tabel 2. 4. *Sequence Diagram*

No	Gambar	Nama	Keterangan
1		<i>Life Line</i>	Objek <i>entity</i> , antar muka yang saling berinteraksi.
2		<i>Message</i>	Komponen ini untuk menggambarkan komunikasi antar objek.
3		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
4		<i>Boundary Class</i>	Menggambarkan penggambaran dari form.
5		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.
6		<i>Control Class</i>	Menggambarkan penghubung antara <i>Boundary</i> dengan tabel.
7		<i>Activation</i>	Sebagai sebuah objek yang akan melakukan sebuah aksi.