

BAB II TINJAUAN PUSTAKA

2.1 Aplikasi

Aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain (Linda, Nursiyanto and Zaini, 2023). Aplikasi dapat mempermudah proses pengolahan data dan meningkatkan layanan kepada pengguna, proses yang fleksibel dan mudah dipahami merupakan keunggulan dari aplikasi (Prasetya and Sudarmilah, 2022).

2.2 Pelayanan

Pelayanan adalah suatu aktivitas atau serangkaian aktivitas yang bersifat tidak kasat mata (tidak dapat diaraba) terjadi sebagai akibat adanya interaksi antara konsumen dengan karyawan atau hal-hal lain yang disediakan oleh perusahaan pemberi pelayanan yang dimaksudkan untuk memecahkan permasalahan konsumen atau pelanggan (Prasetyo, *et al.*, 2022).

Pelayanan administrasi dokumen kapal perikanan mempunyai hal penting untuk melihat prosedur pelayanan administrasi yang telah dilakukan sesuai dengan peraturan yang ada kemudian diambil keputusan apakah pelayanan administrasi dokumen kapal perikanan dan prosedur pelayanan dalam pembuatan persetujuan berlayar kapal perikanan (Agarina *et al.*, 2024).

2.3 Laundry

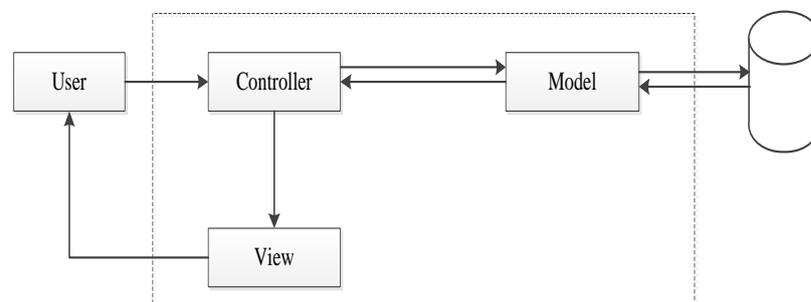
Laundry merupakan salah satu pelayanan jasa di bidang cuci mencuci pakaian dengan memiliki jenis cucian yang telah ditetapkan harga oleh pihak penyedia jasa dan waktu lama cucian biasanya ditentukan oleh penyedia dengan batas minimal dan maksimal selesainya cucian yang dipesan pelanggan (Amnah and Halimah, 2024).

Laundry adalah proses mencuci dan merawat pakaian agar tetap bersih dan terawat. Proses ini melibatkan berbagai langkah, mulai dari memisahkan pakaian

berdasarkan warna dan jenis kain, mencuci dengan deterjen yang sesuai, hingga proses pengeringan dan pelipatan pakaian (Rian and Fuadytama, 2019).

2.4 CodeIgniter

CodeIgniter adalah *Framework* untuk bahasa pemrograman PHP, yang dibuat Rick Ellis pada tahun 2006. *CodeIgniter* memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi secara mudah dan cepat serta memiliki sifat yang fleksibel dapat dikembangkan dalam perangkat web, dekstop maupun mobile. *CodeIgniter* memiliki konsep atau pola *Model-View-Controller* (MVC) sehingga kode-kode dapat di sederhanakan (Raharjo, 2018).



Gambar 2.1 Arsitektur MVC

2.5 PHP

PHP (*Personal Home Page*) adalah pemrograman (*interpreter*) yang melakukan proses penerjemahan baris sumber menjadi kode mesin yang dimengerti oleh komputer secara dinamis (Firdhayanti *et al.*, 2023). Pengertian PHP juga merupakan singkatan dari *Hypertext Preprocessor* dengan Bahasa yang berbentuk skrip yang bersifat *server side* yang dimana proses pengerjaan kode program dilakukan di *server*, dan hasilnya akan ditampilkan di *browser* (Sihombing, 2022).

2.6 MySql

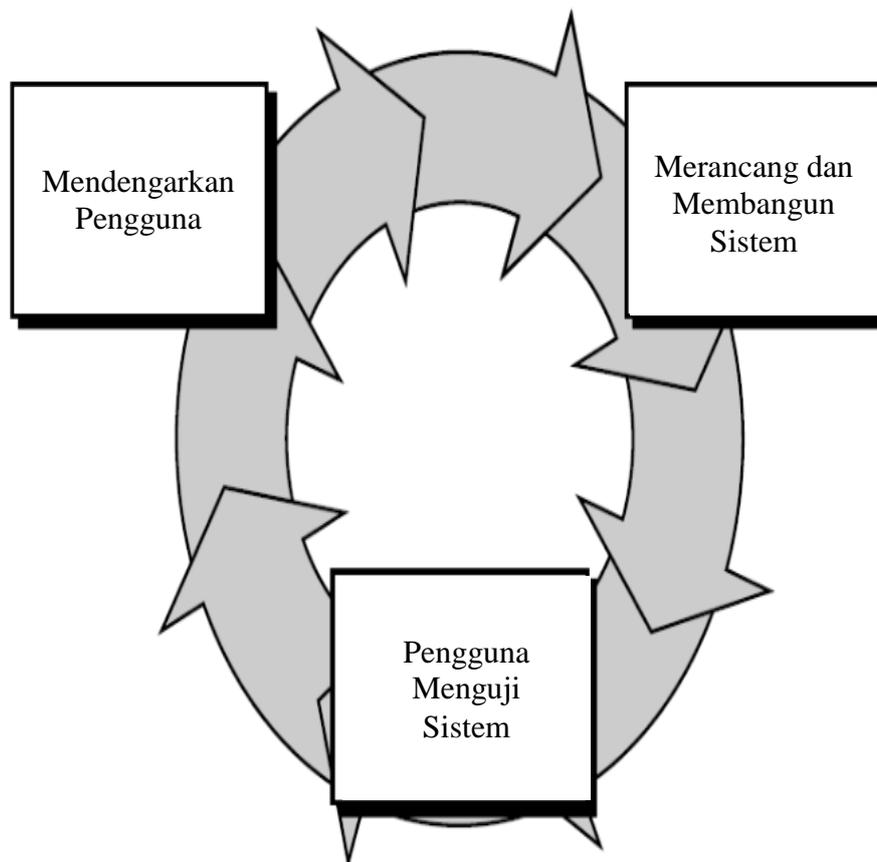
MySQL adalah sebuah *database management system* (manajemen basis data) menggunakan perintah dasar SQL (*Structured Query Language*) yang cukup terkenal. *Database management system* (DBMS) MySQL multi pengguna dan bersifat gratis. Mysql digunakan sebagai wadah dalam mengelola data yang

dapat disimpan digunakan kembali dengan cara yang lebih efisien (Setyawan and Pratiwi, 2019).

2.7 *Prototype*

Prototype merupakan metode yang digunakan untuk mendapatkan kebutuhan pengguna lebih spesifik dalam segi teknis. Model *prototype* dapat digunakan untuk menyambungkan ketidakpahaman pengguna mengenai teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pengguna kepada pengembang perangkat lunak (Rosa dan Shalahuddin, 2019).

Prototype dapat dimulai dengan mengumpulkan kebutuhan pengguna terhadap perangkat lunak yang akan dibuat, kemudian dirancang menggunakan model *prototype* dalam bentuk mockup untuk mempermudah pengguna mengevaluasi *prototype* tersebut (Septiana and Zaini, 2022). Sehingga hasil dari sistem yang dikembangkan telah sesuai dengan permintaan dan kebutuhan pengguna.



Gambar 2.2 Metode *Prototype*

2.7.1 Tahapan Penelitian

Tahapan dalam penelitian sebagai langkah-langkah penelitian yang harus dikerjakan, berikut adalah tahapan penelitian menggunakan *prototype*.

1. Mendengarkan pelanggan

Pelanggan dan pengembang bersama sama mendefinisikan format seluruh perangkat lunak, semua kebutuhan dan garis besar sistem yang akan dibuat.

2. Membangun atau memperbaiki *mockup*

Dengan membuat perancangan sementara seperti *mockup* yang berfokus pada penggambaran terhadap sistem yang dibangun kepada pengguna (Misal membuat rancangan *input* dan *output*). Dalam tahap ini prototyping yang sudah disepakati diterjemahkan kedalam bahasa pemrograman yang sesuai.

3. Pelanggan melihat atau menguji *mockup*

Penggambaran sistem yang digunakan seperti *mockup* perlu di uji kepada pengguna untuk mendapatkan kebutuhan yang sesuai dengan keinginannya, selanjutnya pada sistem yang telah dibangun perlu diuji untuk mengetahui kelayakan terhadap fungsi sistem. Pelanggan akan mengevaluasi apakah perangkat lunak sudah jadi dan sudah sesuai yang diharapkan. Jika Ya maka proses akan dilanjutkan ke tahap selanjutnya. Namun jika Tidak maka mengulang tahap sebelumnya.

2.8 Unified Modelling Language (UML)

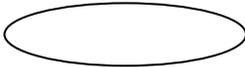
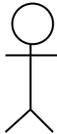
Unified Modelling Language adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Rosa dan Shalahuddin, 2019). Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML.

2.8.1 Use Case Diagram

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Rosa dan Shalahuddin, 2019).

Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada Tabel 2.1.

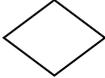
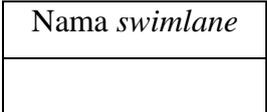
Tabel 2.1 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		<i>Use case</i> Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor Aktor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi/association merupakan komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.		Generalisasi (<i>generalization</i>) merupakan hubungan (umum – khusus) antara dua buah use case yang satu adalah fungsi yang lebih umum
5.	 << Include >>	Include berarti use case yang ditambahkan akan dipanggil saat use case tambahan dijalankan.
6.	 << Extend >>	Ekstensi (<i>extend</i>) merupakan use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.

2.8.2 Activity Diagram

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa dan Shalahuddin, 2019). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada Tabel 2.2:

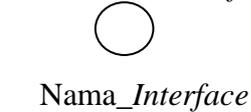
Tabel 2.2 Simbol *Activity Diagram*

No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.

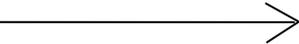
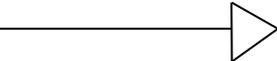
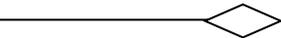
2.8.3 Class Diagram

Class Diagram mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Rosa dan Shalahuddin, 2019). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.		Sama dengan konsep interface dalam pemrograman berorientasi objek.

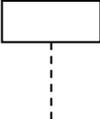
Tabel 2.3 Simbol *Class Diagram* (Lanjutan)

No.	Simbol	Deskripsi
3.	Asosiasi / <i>Asociation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	Asosiasi Berarah / <i>Directed Association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

2.8.1 *Sequence Diagram*

Diagram *Sequence* menjelaskan bagaimana suatu operasi itu dilakukan; message (pesan) apa yang dikirim dan kapan pelaksanaannya (Rosa A.S. dan Shalahuddin, 2019). Berikut simbil *sequence diagram* pada Tabel 2.4.

Tabel 2.4 Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<i>Object lifeline</i> 	Menggambarkan panjang kehidupan suatu objek selama scenario sedang di buat contohnya
2.	<i>Activation</i> 	Dimana proses sedang dilakukan oleh <i>object</i> atau <i>class</i> untuk memenuhi pesan atau perintah

Tabel 2.4 Simbol *Sequence Diagram* (Lanjutan)

No.	Simbol	Deskripsi
3.	<p style="text-align: center;"><i>Message</i></p> 	Sebuah anak panah yang mengindikasikan pesan diantara objek. Dan objek dapat mengirimkan pesan ke dirinya sendiri

2.9 Pengujian *Black Box Testing*

Metode *Black box Testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail *software*. Pengujian *Black Box* memiliki fokus terhadap spesifikasi dari sisi fungsional pada sebuah perangkat lunak. Pada pengujian *Black Box* dengan teknik analisis nilai batas, setiap fungsi yang akan diuji diberikan nilai batas atas serta nilai batas bawah untuk dilihat apakah input dan *output* sudah sesuai maupun tidak. Sering terjadinya kesalahan pada input menjadi salah satu prinsip dari adanya pengujian perangkat lunak dengan teknik analisis nilai batas, di manapada dengan teknik tersebut input akan diuji syarat fungsionalnya (Permatasari *et al.*, 2023).

2.10 Penelitian Terdahulu

Berikut merupakan penelitian terdahulu yang digunakan sebagai acuan atau referensi pada penelitian yang dibuat:

Tabel 2.5 Penelitian Terdahulu

Nama	Judul	Masalah	Metode	Hasil Analisis
(Syahri na and Saptadi , 2022)	Perancangan Sistem Informasi Pada Usaha Laundry	Pelayanan pada Milla laundry saat ini masih dilakukan secara konvensional dan sederhana belum menggunakan sistem yang terkomputerisasi	<i>System Development Life Cycle (SDLC)</i>	Hasil penelitian ini berupa sebuah sistem informasi, dengan adanya sistem informasi jasa laundry ini diharapkan dapat meningkatkan kualitas layanan yang berdampak pada peningkatan pendapatan Milla Laundry.

Tabel 2.5 Penelitian Terdahulu (Lanjutan)

Nama	Judul	Masalah	Metode	Hasil Analisis
(Rafildo, Munti and Azriadi, 2022)	Rancang Bangun Sistem Informasi E-Laundry dengan Implementasi Berbasis Web (Analysis)	Belum tersedianya sistem informasi laundry menggunakan <i>web</i>	<i>Waterfall</i>	Website ini dapat membantu menjalankan bisnis Laundry secara efisien dan memberikan kemudahan baik kepada <i>Costumer</i> untuk bertransaksi dengan pemilik Laundry maupun kepada admin untuk melakukan pengolahan data.
(Prasetyo, Priambodo and Surajyo, 2022)	Sistem Informasi Pelayanan Jasa Laundry Dika	Mendapati kendala dalam pengelolaan data dan transaksi pembayaran. Hal ini disebabkan karena sistem pengelolaan data dan transaksi pembayaran masih secara manual. Sistem	<i>grounded theory</i>	peneliti telah merancang penyimpanan berbasis database yang terintegrasi pada aplikasi agar lebih efektif dan akurat, Pembuatan laporan menggunakan aplikasi sistem baru dan mampu menyajikan serta menyimpan laporan baik data maupun transaksi yang lebih mudah dan akurat dengan sistem yang terinterasi dengan computer serta memunculkan keluhan pengambilan dan pengambilan barang
(Amnah and Halimah, 2024)	Rancang Bangun Aplikasi Antar Jemput Laundry Dengan Metode FIFO Berbasis Android	Sistem pengantaran dan pengambilan laundry mengandalkan komunikasi manual, menyebabkan potensi kesalahan, ketidakjelasan jadwal, dan keterlambatan dalam pelayanan.	<i>Waterfall</i>	enelitian ini juga menggunakan metode FIFO sebagai metode pengatur antrian pesanan agar proses penerimaan pesanan dapat dilakukan secara maksimal. Sehingga dapat memberikan kepuasan yang di rasakan pelanggan karena kepuasan pelanggan merupakan factor penting yang dapat membentuk loyalitas atau loyalitas pelanggan

Tabel 2.5 Penelitian Terdahulu (Lanjutan)

Nama	Judul	Masalah	Metode	Hasil Analisis
(Febriani, Putra and Jakti, 2022)	Implementasi Web Mobile Pada Sistem Informasi Jamaah Dakwah Kota Bandar Lampung	Pelaksanaan pendaftaran dan penjadwalan dilakukan secara manual dengan proses pencatatan dan pelaporan data jamaah sehingga membutuhkan waktu yang lama dan tidak efektif	<i>Extreme Programming</i>	Sistem yang dihasilkan berupa pengolahan data jamaah, pendaftaran, agenda, informasi kegiatan dan laporan berbasis <i>web mobile</i> dengan menggunakan bahasa PHP dan <i>database Mysql</i> dan <i>tools editor Dreamweaver</i> dan <i>Sqlyoq</i> .