

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Penelitian bertajuk “Pemanfaatan *Smartphone* untuk Mengajarkan Anak Menggunakan Folder Pakaian” dilakukan oleh Ilham Saputra (2020). Hasil penelitian yang diperoleh antara lain: alat pelipat pakaian dapat dikontrol melalui *smartphone*, motor *servo* dapat melipat pakaian tertentu, informasi dapat ditampilkan pada *LCD*, modul *Bluetooth* memungkinkan alat dapat dihubungkan dengan *smartphone*, dan speaker dapat menghasilkan suara[2].

Erwin Sukma Bukardi melakukan penelitian “Perancangan dan Pembuatan Mesin Pembungkus Kaos Semi Otomatis Menggunakan Metode *Fuzziness Proportional Derivative* (fpd)” pada tahun 2015. Hasil penelitian menunjukkan bahwa mekanisme Mesin Lipat Kaos terdiri dari empat lengan lipat, atau folder lipat, terhubung ke motor DC. Mesin tersebut memanfaatkan metode turunan proporsional *fuzzy* (FPD) dengan dua *input kesalahan delta* (ΔErr) dan *kesalahan* (*Err*), serta satu siklus pelipatan membutuhkan waktu 9,56 detik[3].

Amirah (2021) melakukan penelitian untuk proyeknya yang diberi nama "Desain Alat Lipat Pakaian Berbasis *Android*". Berdasarkan temuan penelitian, alat pelipat pakaian berbasis *Arduino Uno* ini mampu melipat pakaian dan kaos dewasa dengan ukuran S, M, L, dan XL. Pasalnya,

penggunaan alat lipat seringkali hanya memakan waktu ± 6 detik dibandingkan ± 10 detik saat melipat[4].

Proyek penelitian Natasya Clariza Rostanti bertajuk “Aplikasi berbasis web untuk mengelola pencatatan transaksi jual beli pakaian jadi (studi kasus: *yes no limite*, Salatiga)” (2021). Berdasarkan temuan penelitian, pelaku usaha *laundry* dapat menggunakan aplikasi ini untuk mengelola penjualan, pembelian, dan pembayaran biaya operasional. Mampu menghasilkan laporan laba rugi, daftar stok, buku besar, laporan penjualan, dan jurnal umum selain catatan akuntansi[5].

Proyek penelitian Elsa Oktavia tahun 2020 berjudul Pembangunan Sistem Informasi Industri Jasa Jahit *Online Berbasis Web* Dengan Pendekatan *Waterfall*. Temuan penelitian ini memudahkan pengelolaan data pesanan klien oleh administrator dalam program *web* yang terkomputerisasi. Prosedur transaksi lebih fleksibel jika nasabah menggunakan aplikasi *online* karena nasabah dapat melakukan transaksi dari mana saja[6].

2.2 Landasan Teori

2.2.1 PHP

Biasanya, *PHP* (*Hypertext Preprocessor*), bahasa pemrograman sumber terbuka, digunakan untuk mengembangkan situs web yang dinamis dan interaktif. *PHP* dapat digunakan di *server* web bersama dengan *HTML*, *CSS*, dan *JavaScript* untuk membuat halaman web dinamis[7]. Ini ditunjukkan pada gambar 2.1.



Gambar 2.1 *PHP*

(Sumber: <https://images.app.goo.gl/LYTFhDroeDyBBPzr7>)

2.2.2 Sistem Informasi

Integrasi teknologi informasi dengan perilaku manusia untuk mendukung operasional dan manajemen dikenal dengan istilah sistem informasi. Ungkapan "sistem informasi", bila digunakan secara umum, sering kali mengacu pada interaksi yang terjadi antara manusia, teknologi, data, dan proses *algoritmik*.

2.2.3 *MySQL*

Sistem manajemen basis data relasional (*RDBMS*) tersedia dalam sumber terbuka berdasarkan *SQL* disebut *MySQL*. *RDBMS* merupakan suatu sistem pengelolaan basis data yang didasarkan pada data-data yang saling berkaitan (memiliki hubungan) di dalam satu kelompok, jika *DBMS* merupakan suatu sistem pengelolaan basis data pada[8]. Ini ditunjukkan pada gambar 2.2.



Gambar 2.2 *MYSQL*

(Sumber: <https://images.app.goo.gl/LYTFhDroeDyBBPzr7>)

2.2.4 *Visual Studio Code*

Editor teks yang dapat diandalkan dan ringan adalah *Visual Studio Code*. *Microsoft* mengembangkan editor teks ini khusus untuk sistem operasi multiplatform. Bahasa pemrograman *Java Script*, *Node.js*, dan *TypeScript* didukung oleh *Visual Studio Code*[9]. Ini ditunjukkan pada gambar 2.3.



Gambar 2.3 *Visual Studio Code*

(Sumber: <https://images.app.goo.gl/Fd3BM8xaHpimSFhi8>)

2.2.5 *Database*

Informasi dalam *database* adalah pengelompokan data terhubung yang disimpan dalam sistem tertentu untuk memfasilitasi pengelolaan data [10].

2.2.6 *UML (Unified Modelling Language)*

Metodologi *RPL* (Rekayasa Perangkat Lunak) menggunakan alat yang disebut *Unified Modeling Language*, atau disingkat *UML*, untuk menggambarkan tujuan, fungsi, aliran, dan mekanisme kontrol suatu sistem.

Empat model *UML* yang paling sering digunakan untuk menjelaskan desain sistem adalah diagram urutan, diagram mesin

keadaan perilaku, diagram kelas, dan diagram *usecase*. Keempat teknik dasar tersebut merupakan nama lain dari teknik pemodelan tersebut dalam *Unified Modeling Language*. Keempat pendekatan *UML* ini adalah yang biasanya digunakan dalam proyek yang berorientasi objek[11]. Ini ditunjukkan pada gambar 2.4.



Gambar 2.4 *UML*

(Sumber: https://en.wikipedia.org/wiki/Unified_Modeling_Language)


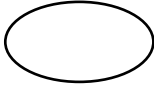


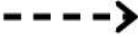
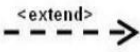
Dalam perancangan sistem terdapat Penggunaan *UML* berikut ini umum terjadi:

1. *Usecase Diagram*

Diagram *use case* merupakan alat yang digunakan dalam pembuatan sistem informasi untuk mensimulasikan perilaku sistem. Selain itu, diagram ini bersifat statis. Berikut merupakan simbol-simbol dari *UseCase Diagram* dapat dilihat pada Tabel 2.1.

Tabel 2. 1 *Usecase Diagram*




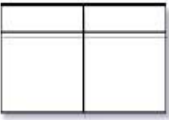
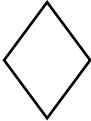

Simbol	Nama	Detail
--------	------	--------

	Penampil	Fungsi administrator dalam kaitannya dengan <i>use case</i> .
	<i>Usecase</i>	Gangguan dan komunikasi antara aktor dan sistem.
	<i>Generalisation</i>	Sebagai titik penghubung antara dua objek.
	<i>System</i>	Pilih paket yang hanya memberikan pandangan terbatas pada sistem.
	<i>Include</i>	Perjelas kasus penggunaan sumber tersebut.
	<i>Extend</i>	menunjukkan bahwa, jika suatu kondisi terpenuhi, satu Penambahan fungsional pada use case lain disebut use case.

2. *Activity Diagram*

Diagram aktivitas, kadang-kadang disebut diagram aktivitas, adalah diagram statis yang menunjukkan bagaimana sistem bisnis beroperasi. Diagram dapat dilihat pada Tabel 2.2.

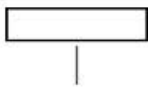
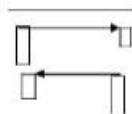







Tabel 2. 2 *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Action</i>	Status sistem setelah selesainya tindakan.
	<i>Initial Node</i>	Bagaimana segala sesuatunya berasal atau terbentuk.
	<i>Initial Final Node</i>	Bagaimana segala sesuatunya bermula dan berakhir.
	<i>Vertical Swimlans</i>	Diagram alur proses yang menjelaskan bagaimana berbagai komponen proses berinteraksi.
	<i>Decision</i>	Tentukan kriteria pengujian untuk memastikan bahwa objek atau aliran kontrol lebih sering mengikuti satu jalur.
	<i>Control Flow</i>	Dalam alur kerja, tindakan tertentu berfungsi sebagai sarana untuk mengekspresikan kendali atas suatu aktivitas.

3. *Sequence Diagram*

Diagram urutan, sering dikenal sebagai diagram interaksi, menunjukkan bagaimana pesan dikirim dan diterima antar objek. Tabel menampilkan simbol-simbol yang digunakan dalam diagram urutan. Diagram dapat dilihat pada Tabel 2.3.

Tabel 2. 3 *Sequence Diagram*

Simbol	Nama	Keterangan
	<i>Lifeline</i>	Antarmuka dengan interaksi timbal balik, atau objek entitas.
	<i>Message</i>	Spesifikasi komunikasi antar item yang menyimpan data tentang peristiwa yang terjadi.
	<i>Actor</i>	menggambarkan bagaimana pengguna terlibat dengan sistem.
	<i>Boundary Class</i>	menjelaskan bagaimana formulir ditampilkan.
	<i>Entity Class</i>	Jelaskan hubungan antara tugas-tugas yang perlu diselesaikan.
	<i>Control Class</i>	Jelaskan hubungan antara tabel dan batas.
	<i>Activation</i>	Sebagai sesuatu yang akan melaksanakan suatu tugas.
	<i>Message</i>	menunjukkan bahwa item berkomunikasi satu sama lain.
	<i>Self Message</i>	menunjukkan kembalinya komunikasi ke dalam benda itu sendiri.