

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian ini berjudul “Sistem Pemesanan Print Textile Dengan Model *Finite State Machine*” yang akan diimplementasikan pada Mochi Studio. Maka dalam tinjauan pustaka ini berisi teori-teori dasar dalam perancangan sistem ini meliputi teori tentang konsep sistem pemesanan online, *finite state machine*, UML serta *Database*.

2.1.1. *Finite State Machine* (FSM)

Finite State Machine (FSM) adalah model pengendalian sistem yang digunakan untuk menggambarkan perilaku, yakni apa yang harus dilakukan dalam semua kemungkinan situasi. FSM memperkenalkan konsep state sebagai informasi tentang sejarah sistem di masa lalu. Semua state mewakili situasi-situasi berbeda yang mungkin terjadi dalam suatu sistem, dan karena itu FSM memiliki memori terhadap bagaimana sistem bisa mencapai kondisi saat ini[7]. Konsepnya sederhana namun sangat efektif untuk mengelola kompleksitas alur kerja yang berurutan. FSM dibangun dari tiga komponen dasar yang saling berinteraksi:

1. *State* (Keadaan): Merupakan kondisi unik di mana sistem dapat berada pada suatu waktu. Setiap *state* merepresentasikan tahapan tertentu dalam proses. Misalnya, dalam sistem pemesanan, state bisa berupa "Pesanan Dibuat", "Menunggu Pembayaran", "Diproses", "Selesai Dicitak", atau "Dikirim". Jumlah *state* dalam FSM harus terbatas.
2. *Input* (Masukan): Adalah peristiwa atau data eksternal yang memicu perubahan *state*. *Input* bertindak sebagai pemicu transisi. Contoh *input* dalam sistem pemesanan bisa berupa "Pelanggan Melakukan Pembayaran", "Desain Disetujui", "Proses Cetak Selesai", atau "Pengiriman Berhasil".
3. *Transition* (Transisi): Adalah perubahan dari satu *state* ke *state* lain sebagai respon terhadap *input* tertentu. Setiap *transition* memiliki *state* asal, *state* tujuan, dan *input* yang memicu transisi tersebut, serta mungkin kondisi tambahan atau aksi yang dilakukan saat transisi terjadi. Transisi ini mendefinisikan alur logis dari proses.

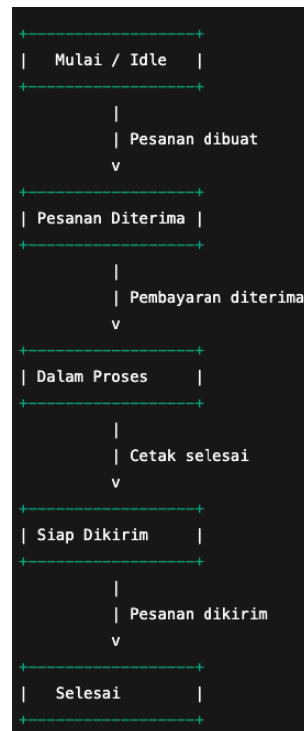
FSM memiliki implementasi yang lebih luas dalam berbagai sistem informasi, antara lain:

- a. Pengembangan *Game*: Mengontrol perilaku karakter non-pemain (NPC) dan alur cerita.
- b. Protokol Jaringan: Memodelkan status koneksi dan pertukaran pesan antar perangkat.
- c. Sistem Kontrol: Mengelola status mesin atau perangkat otomatis (misalnya, mesin penjual otomatis, lampu lalu lintas).
- d. Manajemen Alur Kerja (*Workflow Management*): Memodelkan langkah-langkah dalam proses bisnis, seperti persetujuan dokumen, manajemen pesanan, atau siklus hidup produk.
- e. Sistem Pembayaran: Mengelola berbagai status transaksi pembayaran (menunggu, berhasil, gagal, dibatalkan, dikembalikan).

Penggunaan FSM dalam sistem pemesanan print kain di Mochi Studio memiliki beberapa keuntungan signifikan:

- a. Kejelasan Alur Proses: FSM menyediakan representasi visual yang jelas dan terstruktur dari setiap tahapan pemesanan (dari pembuatan pemesanan hingga pengiriman), memudahkan pemahaman dan komunikasi antar tim.
- b. Konsistensi dan Validasi: Memastikan bahwa setiap pesanan hanya dapat berpindah dari satu *state* ke *state* lain jika semua kondisi yang ditentukan terpenuhi, sehingga mencegah *state* yang tidak valid atau urutan proses yang salah.
- c. Penanganan Kompleksitas: Sistem pemesanan print kain melibatkan banyak *state* dan input yang berbeda (desain, pembayaran, produksi, pengiriman). FSM membantu mengelola kompleksitas ini secara modular dan terorganisir.
- d. Skalabilitas dan Pemeliharaan: Dengan FSM, logika bisnis terisolasi dalam *state* dan *transition*, membuatnya lebih mudah untuk menambah *state* baru, mengubah alur, atau memperbaiki *bug* tanpa mempengaruhi bagian lain dari sistem.
- e. Pelacakan Status *Real-time*: Memungkinkan Mochi Studio untuk melacak status setiap pesanan secara *real-time*, memberikan informasi akurat kepada pelanggan dan internal.

Model *Finite State Machine* dapat digambarkan dalam bentuk graf yang menunjukkan *state* (keadaan) dan *transition* (perpindahan antar *state*) berdasarkan input tertentu. Setiap node pada graf merepresentasikan suatu *state*, sedangkan panah (arah) menunjukkan transisi antar *state*. Berikut contoh model umum FSM :



Gambar 2. 1 Model Umum FSM

2.1.2 Sistem Pemesanan Online

Sistem pemesanan online adalah aplikasi berbasis *web* atau *mobile* yang memungkinkan pelanggan untuk memesan barang atau jasa melalui internet tanpa perlu interaksi fisik langsung. Sistem ini memfasilitasi seluruh proses dari pemilihan produk, penyesuaian (jika ada), pembayaran, hingga konfirmasi pesanan dan pelacakan pengiriman. Tujuannya adalah memberikan kemudahan, kecepatan, dan aksesibilitas bagi pelanggan, serta efisiensi operasional bagi penyedia layanan.

2.1.3 Teknologi *Web Development*

Website adalah kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara, dan video atau gabungan dari semuanya yang disediakan melalui jalur

koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia. Halaman website dibuat menggunakan bahasa standar yaitu *Hyper Text Markup Language* (HTML). Skrip HTML ini akan diterjemahkan oleh *web browser* sehingga dapat dibaca oleh semua orang[3].

2.1.4 UML

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik / gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object Oriented*). UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti JAVA, C , Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah *object-oriented database Unified Modeling Language* (UML) adalah sebuah bahasa 19 pemodelan perangkat lunak yang telah distandarisasi sebagai media penulisan cetak biru (*blueprints*) perangkat lunak. Ada beberapa jenis UML yang umum digunakan :

A. Use Case Diagram

Diagram *Use Case* digunakan untuk menggambarkan hubungan antar aktor yang terlibat dalam sistem.

B. Activity Diagram

Diagram *Activity* digunakan untuk menggambarkan aliran kerja atau aktivitas dalam sistem.

C. Sequence Diagram

Diagram *Sequence* menggambarkan interaksi antar objek dalam sistem secara berurutan.

D. Class Diagram

Diagram *Class* menggambarkan kelas-kelas yang terdapat dalam sistem dan hubungan antar kelas satu dengan yang lainnya.

2.1.5 ERD

Entity Relationship Diagram (ERD) adalah alat pemodelan data konseptual yang digunakan untuk mengidentifikasi, menyimpan, dan mengambil kembali data sesuai kebutuhan sistem informasi. ERD mempermudah analisis terhadap kebutuhan data serta menggambarkan struktur hubungan antar entitas dalam *database*, yang mencakup entitas, atribut, dan relasi antar elemen data[5]. ERD mempunyai komponen utama berupa :

a. Entitas

Sebuah fokus dalam ERD berupa objek yang dapat terdiri dari beberapa atribut seperti pelanggan, barang dan sebagainya.

b. Atribut

Atribut adalah informasi dalam entitas tersebut seperti *nullable*, *primary key*, *foreign key* dan banyak jenis lainnya.

c. Relasi

Relasi adalah hubungan antar entitas dimana bisa beberapa macam

1. *One to One* : hubungan dimana satu entitas hanya bisa melekat pada 1 entitas lain.
2. *One to Many* : hubungan dimana 1 entitas dapat melekat pada beberapa entitas sekaligus.
3. *Many to Many* : hubungan dimana beberapa entitas juga merupakan nilai dari entitas lainnya.

2.1.6 Database

Basis Data adalah kumpulan data yang terorganisir, disimpan dan dapat diakses secara elektronik dari sebuah komputer. Fungsi dari *Database* adalah memudahkan identifikasi data. Menurut CJ Date, *Database* adalah koleksi “data operasional” yang tersimpan dan juga dipakai oleh sistem aplikasi dari suatu organisasi.

- a. *Data input* adalah data yang masuk dari luar sistem.
- b. *Data output* adalah data yang dihasilkan oleh sistem.
- c. *Data operasional* adalah data yang tersimpan pada sistem.

Database adalah koleksi atau kumpulan data yang mekanis, terbagi, terdefinisi secara formal dan juga dikontrol terpusat pada suatu organisasi (menurut Gordon C. Everest). Dari sumber diatas, *Database* adalah kumpulan dari banyak data yang saling terkait dan terkumpul dalam satu tempat yang sama dan dipakai oleh sistem aplikasi yang dikontrol secara terpusat memiliki serta memiliki nilai yang berharga bagi pemilik[4]. Dalam penelitian kali ini penulis menggunakan basis data berjenis SQL untuk diterapkan dalam sistem.

2.2 Penelitian Terkait

1. Agustin et al. (2021) mengimplementasikan metode *Finite State Machine* (FSM) pada permainan tradisional Setatak berbasis Android. Tujuan utamanya adalah untuk melestarikan budaya lokal melalui media digital yang interaktif dan edukatif bagi anak-

anak. FSM dipilih karena mampu mengatur logika dan alur perilaku sistem *game* secara sistematis, terutama dalam menggambarkan perpindahan pemain dari satu petak ke petak lainnya selama permainan berlangsung. Metode FSM yang digunakan menggambarkan prinsip kerja sistem melalui tiga komponen utama, yaitu *State* (keadaan), *Event* (kejadian), dan *Action* (aksi). Dengan pendekatan ini, sistem permainan dapat mengatur transisi antar status secara logis dan terkendali [2].

2. Penelitian oleh Pratama et al.(2025) merancang dan mengembangkan sistem informasi pemesanan digital printing berbasis *web* untuk mendukung efisiensi layanan di Medina Printing. Sebelum sistem diterapkan, proses pemesanan masih dilakukan secara konvensional dan mengharuskan pelanggan datang langsung ke lokasi, yang dianggap kurang efektif dan berdampak pada menurunnya minat pelanggan. Untuk mengatasi hal ini, sistem dirancang dengan fitur pemesanan *online*, pembayaran digital, serta unggah desain, yang diimplementasikan menggunakan metode *Rapid Application Development* (RAD). Sistem ini bertujuan untuk mempercepat transaksi, meningkatkan kemudahan akses layanan, dan memperluas jangkauan pelanggan. Hasil pengujian menunjukkan bahwa sistem berhasil meningkatkan kualitas layanan dan memberikan kemudahan dalam proses pemesanan secara digital [1].
3. Patria (2021) mengembangkan sebuah aplikasi pemesanan *fitting* pakaian berbasis web untuk Toko Nabeel Textile sebagai respons terhadap kebutuhan layanan daring di masa pandemi. Aplikasi ini dirancang untuk memudahkan pelanggan dalam menjadwalkan sesi *fitting* busana tanpa harus datang langsung ke toko. Melalui sistem ini, pelanggan dapat memilih produk, menentukan waktu *fitting*, serta melihat ketersediaan layanan melalui perangkat digital seperti smartphone dan laptop. Sebelumnya, proses pemesanan masih dilakukan secara manual dan pencatatan hanya menggunakan file Excel, yang dinilai kurang efisien dan rentan terhadap kesalahan pencatatan. Sistem dikembangkan menggunakan metode *Waterfall* dengan teknologi berbasis *web* seperti HTML, PHP, dan *framework* CodeIgniter. Hasil dari pengembangan aplikasi menunjukkan bahwa proses pelayanan menjadi lebih efektif dan efisien, baik dari sisi pelanggan maupun pengelola toko [6].