

# **SISTEM PEMESANAN PRINT TEKSTIL DENGAN MODEL FINITE STATE MACHINE**



## **TUGAS AKHIR**

Diajukan untuk Memenuhi Salah Satu Syarat Kelulusan Program Sarjana

**Disusun oleh:**

**JAISY AL HUDA SHIBGHOTULLOH**

**12240135**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS NUSA MANDIRI**

**JAKARTA**

**2025**

## **LEMBAR PERSEMBAHAN**

Dengan mengucap puji syukur kepada Allah S.W.T, Tugas Akhir ini kupersembahkan untuk:

**(Alm) Abi, (Alm) Umi serta Istri saya tercinta**

Atas segala doa, cinta, pengorbanan, dan dukungan moril maupun spiritual yang tak pernah putus. Terima kasih telah menjadi kekuatan utama dalam setiap langkah penulis.

**Saudara-saudaraku tercinta**

Yang selalu memberikan semangat, canda, dan motivasi dalam proses perjalanan ini.

**Dosen pembimbing dan seluruh dosen Universitas Nusa Mandiri**

Atas ilmu, arahan, dan bimbingan yang telah diberikan selama masa perkuliahan hingga proses penyusunan Tugas Akhir ini.

**Rekan-rekan seperjuangan kelas 12.8C.01**

Yang telah menemani dalam suka dan duka, berbagi ilmu, tawa, dan semangat dalam setiap proses belajar.

**Seluruh pihak yang turut membantu**

Yang tidak dapat penulis sebutkan satu per satu, namun peran dan bantuannya sangat berarti bagi terselesaikannya karya ini.

Semoga Tugas Akhir ini dapat menjadi awal langkah untuk memberikan kontribusi nyata di bidang teknologi informasi dan menjadi amal jariyah bagi semua yang terlibat.

## LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Jaisy Al Huda Shibghotulloh  
NIM : 12240135  
Program Studi : Informatika  
Fakultas : Teknologi Informasi  
Perguruan Tinggi : Universitas Nusa Mandiri

Dengan ini menyatakan bahwa Tugas Akhir yang telah saya buat dengan judul: **“Sistem Pemesanan Print Tekstil Dengan Model Finite State Machine”** adalah benar-benar hasil karya saya sendiri, orisinal (asli) dan tidak mengandung unsur plagiat, serta belum pernah diterbitkan atau dipublikasikan dalam bentuk dan tempat mana pun.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya tanpa ada paksaan dari pihak mana pun. Apabila di kemudian hari terdapat bukti bahwa karya ini merupakan hasil plagiat atau diklaim sebagai milik orang lain atau lembaga tertentu, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku, termasuk pencabutan kelulusan dari Universitas Nusa Mandiri..

Dibuat di : Jakarta

Pada tanggal : 28 Juli 2025

Yang menyatakan,



Jaisy Al Huda Shibghotulloh

## LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA

### ILMIAH

Yang bertanda tangan di bawah ini:

Nama : Jaisy Al Huda Shibghotulloh  
NIM : 12240135  
Program Studi : Informatika  
Fakultas : Teknologi Informasi  
Perguruan Tinggi : Universitas Nusa Mandiri

**Mandiri**, berupa Hak Bebas Royalti Non-Eksklusif (Non-exclusive Royalty-Free Right) atas karya ilmiah saya yang berjudul: **“Sistem Pemesanan Print Tekstil Dengan Model Finite State Machine”**, beserta perangkat yang diperlukan (apabila ada).

Dengan Hak Bebas Royalti Non-Eksklusif ini, pihak Universitas Nusa Mandiri berhak menyimpan, mengalih-media atau mengubah format, mengelola dalam pangkalan data (database), mendistribusikan, serta menampilkan atau mempublikasikan karya ilmiah saya melalui internet atau media lainnya untuk kepentingan akademis, tanpa perlu meminta izin kembali selama tetap mencantumkan nama saya sebagai penulis/pencipta karya ilmiah tersebut.

Saya menyatakan bersedia menanggung secara pribadi, tanpa melibatkan pihak Universitas Nusa Mandiri, atas segala bentuk tuntutan hukum yang mungkin timbul terkait pelanggaran Hak Cipta atas karya ilmiah ini.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya untuk dapat digunakan sebagaimana mestinya.

Dibuat di : Jakarta

Pada tanggal : 28 Juli 2025

Yang menyatakan,



**Jaisy Al Huda Shibghotulloh**



## LEMBAR PERSETUJUAN DAN PENGESAHAN TUGAS AKHIR

Tugas Akhir ini diajukan oleh:

Nama : Jaisy Al Huda Shibghotulloh  
NIM : 12240135  
Program Studi : Informatika  
Fakultas : Teknologi Informasi  
Jenjang : Strata Satu (S1)  
Judul Tugas Akhir : **Sistem Pemesanan Print Tekstil Dengan Model Finite State Machine**

Untuk dipertahankan pada Periode I-2025 dihadapan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh Sarjana Komputer (S.Kom) pada Program Sarjana Program Studi Informatika Fakultas Teknologi Informasi di Universitas Nusa Mandiri.

Jakarta, 4 Agustus 2025

### PEMBIMBING TUGAS AKHIR

Dosen Pembimbing : Astriana Mulyani, S.Si., M.Kom



### DEWAN PENGUJI

Penguji I :

Linda Marlinda, M.Kom.



Penguji II :

Sita Anggraeni, M.Kom.



## LEMBAR PANDUAN PENGGUNAAN HAK CIPTA

Tugas Akhir yang berjudul “**Sistem Pemesanan Print Tekstil Dengan Model Finite State Machine**” adalah hasil karya tulis asli Jaisy Al Huda Shibghotulloh dan bukan hasil terbitan sehingga peredaran karya tulis hanya berlaku di lingkungan akademik saja, serta memiliki hak cipta. Oleh karena itu, dilarang keras untuk menggandakan baik sebagian maupun seluruhnya karya tulis ini, tanpa seizin penulis.

Referensi kepustakaan diperkenankan untuk dicatat tetapi pengutipan atau peringkasan isi tulisan hanya dapat dilakukan dengan seizin penulis dan disertai ketentuan pengutipan secara ilmiah dengan menyebutkan sumbernya.

Untuk keperluan perizinan pada pemilik dapat menghubungi informasi yang tertera di bawah ini:

Nama : Jaisy Al Huda Shibghotulloh  
Alamat : Jl. Arjuna no 113. Kec. Tegal Timur, Kota Tegal, Provinsi Jawa Tengah  
No.Telp : 0858-6907-0071  
E-mail : jalzae@gmail.com

## KATA PENGANTAR

Alhamdulillahirabbil'alamin, segala puji syukur penulis panjatkan ke hadirat Allah SWT atas segala limpahan rahmat, taufik, dan hidayah-Nya. Berkat izin dan kehendak-Nya, penulis akhirnya dapat menyelesaikan penyusunan Tugas Akhir ini dengan judul:

### **“SISTEM PEMESANAN PRINT TEKSTIL DENGAN MODEL FINITE STATE MACHINE.”**

Tugas Akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Nusa Mandiri. Dalam proses penyusunan Tugas Akhir ini, penulis banyak mendapatkan dukungan, bimbingan, serta bantuan dari berbagai pihak yang tak ternilai harganya.

Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih dan penghargaan setinggi-tingginya kepada:

1. Allah Subhanahu Wa Ta'ala atas segala nikmat dan kemudahan yang telah dilimpahkan pada saya
2. Rektor Universitas Nusa Mandiri
3. Wakil Rektor I Bidang Akademik Universitas Nusa Mandiri
4. Dekan Fakultas Teknologi Informasi Universitas Nusa Mandiri
5. Ketua Program Studi Informatika Universitas Nusa Mandiri
6. Ibu **Astriana Mulyani, S.Si, M.Kom**, selaku Dosen Pembimbing Tugas Akhir yang telah memberikan bimbingan, masukan, dan arahan secara sabar dan konsisten
7. Seluruh Bapak/Ibu Dosen Program Studi Informatika Universitas Nusa Mandiri yang telah berbagi ilmu dan pengalaman selama masa studi
8. Staff dan karyawan Fakultas Teknologi Informasi yang telah membantu proses administrasi perkuliahan

9. Kedua orang tua tercinta, istri dan juga sanak saudara saya atas segala doa, kasih sayang, dan dukungan moril maupun spiritual yang tidak pernah berhenti
10. Seluruh rekan-rekan mahasiswa kelas 12.8C.01 atas kebersamaan, semangat, dan saling mendukung selama masa kuliah
11. Semua pihak yang tidak dapat disebutkan satu per satu yang telah membantu baik secara langsung maupun tidak langsung dalam penyelesaian Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa yang akan datang.

Akhir kata, semoga Tugas Akhir ini dapat memberikan manfaat, baik bagi penulis sendiri maupun bagi pembaca yang memiliki minat di bidang pengembangan sistem informasi, khususnya dalam implementasi aplikasi manajemen laundry berbasis web.

**Jakarta, 28 Juli 2025**

Penulis

**Jaisy Al Huda Shibhotulloh**

## DAFTAR ISI

LEMBAR JUDUL .....	I
LEMBAR PERSEMBAHAN.....	II
LEMBAR PANDUAN PENGGUNAAN HAK CIPTA.....	VI
KATA PENGANTAR .....	VII
DAFTAR ISI .....	IX
DAFTAR GAMBAR .....	XI
DAFTAR TABEL .....	XII
ABSTRAK .....	XIII
ABSTRACT.....	XIV
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang Masalah .....	1
1.2. Identifikasi Permasalahan .....	3
1.3. Perumusan Masalah .....	3
1.4. Maksud dan Tujuan.....	3
1.5. Metode Penelitian .....	4
1.6. Ruang Lingkup.....	7
<b>BAB II LANDASAN TEORI.....</b>	<b>8</b>
2.1. Tinjauan Pustaka.....	8
2.2. Penelitian Terkait .....	12
<b>BAB III ANALISA SISTEM BERJALAN.....</b>	<b>14</b>
3.1. Tinjauan Institusi/Perusahaan .....	14
3.2. Proses Bisnis .....	15
3.3. Spesifikasi Dokumen Sistem Berjalan.....	17
<b>BAB IV RANCANGAN SISTEM DAN PROGRAM USULAN .....</b>	<b>18</b>
4.1. Analisa Kebutuhan Software .....	18
4.2. Desain .....	21
4.3. <i>Code Generation</i> .....	48
4.4. <i>Testing</i> .....	52
4.5. <i>Support</i> .....	54

4.6. Spesifikasi Dokumen Sistem Usulan .....	57
<b>BAB V PENUTUP .....</b>	<b>62</b>
5.1. Kesimpulan .....	62
5.2. Saran .....	63
<b>DAFTAR PUSTAKA.....</b>	<b>64</b>
<b>DAFTAR RIWAYAT HIDUP .....</b>	<b>65</b>
<b>LAMPIRAN .....</b>	<b>68</b>

## DAFTAR GAMBAR

Gambar 2. 1 Model Umum FSM .....	10
Gambar 3. 1 Proses Bisnis .....	16
Gambar 4. 1 Use Case Diagram.....	23
Gambar 4. 2 Class Diagram .....	24
Gambar 4. 3 Sequence Diagram Pendaftaran .....	26
Gambar 4. 4 Sequence Diagram Login.....	26
Gambar 4. 5 Sequence Diagram Checkout.....	27
Gambar 4. 6 Sequence Diagram Pemesanan .....	27
Gambar 4. 7 Sequence Diagram Laporan Pesanan.....	28
Gambar 4. 8 Component Diagram.....	29
Gambar 4. 9 Deployment Diagram.....	29
Gambar 4. 10 Ilustrasi Logika Entitas .....	31
Gambar 4. 11 Halaman Registrasi .....	40
Gambar 4. 12 Halaman Login.....	41
Gambar 4. 13 Halaman Beranda Pelanggan .....	41
Gambar 4. 14 Halaman Beranda Admin.....	42
Gambar 4. 15 Halaman Produk.....	42
Gambar 4. 16 Halaman Detail Produk.....	43
Gambar 4. 17 Halaman Pemesanan .....	43
Gambar 4. 18 Halaman Laporan Pemesanan Admin.....	44
Gambar 4. 19 Struktur Express Js.....	49
Gambar 4. 20 Struktur NuxtJs .....	50
Gambar 4. 21 Publikasi Web di Netlify.....	54



## DAFTAR TABEL

Tabel IV. 1 Tabel Halaman Website.....	38
Tabel IV. 2 Tabel Spesifikasi.....	56

## ABSTRAK

### **Jaisy Al Huda Shibghotulloh (12240135), SISTEM PEMESANAN PRINT TEXTILE DENGAN MODEL FINITE STATE MACHINE**

Pengelolaan pemesanan pada usaha print tekstil seperti Mochi Studio sebelumnya masih dilakukan secara manual melalui pesan instan, yang menyebabkan alur kerja tidak terdokumentasi dengan baik, rentan kesalahan, serta sulit dipantau secara *real-time*. Penelitian ini bertujuan untuk merancang dan membangun sistem pemesanan digital berbasis web yang memanfaatkan pendekatan *Finite State Machine* (FSM) guna memastikan alur proses yang sistematis dan terkendali pada setiap tahapan pesanan. Sistem dikembangkan menggunakan framework Express.js sebagai *backend* dan Nuxt.js pada sisi *frontend*, serta ditulis dengan bahasa TypeScript untuk meningkatkan konsistensi dan keandalan kode. Metodologi pengembangan yang digunakan bersifat iteratif untuk memungkinkan penyesuaian cepat terhadap kebutuhan pengguna. Fitur utama dalam sistem ini meliputi pendaftaran dan login pengguna, pembuatan pesanan, pelacakan status produksi berbasis FSM, integrasi pelacakan pengiriman (*airwaybill*), serta pengelolaan data pelanggan. Hasil implementasi menunjukkan bahwa sistem mampu menjalankan seluruh fungsi utama dengan baik dan memberikan pengalaman pemesanan yang lebih terstruktur, efisien, dan mudah dipantau oleh pelanggan maupun pihak pengelola. Dengan demikian, sistem ini diharapkan dapat meningkatkan profesionalisme layanan serta mendukung pengambilan keputusan berbasis data di bidang industri kreatif tekstil.

**Kata Kunci:** *Finite State Machine*, Node Js, TypeScript, Pemesanan Print Tekstil, Sistem Informasi Web

## ABSTRACT

**Jaisy Al Huda Shibhotulloh (12240135), *TEXTILE PRINT ORDERING SYSTEM USING FINITE STATE MACHINE MODEL***

*Order management in textile printing businesses such as Mochi Studio was previously carried out manually via instant messaging, leading to poorly documented workflows, a high risk of errors, and difficulties in real-time monitoring. This study aims to design and develop a web-based digital ordering system utilizing the Finite State Machine (FSM) approach to ensure a systematic and controlled process flow at each stage of the order lifecycle. The system was developed using the Express.js framework for the backend and Nuxt.js for the frontend, and it is written in TypeScript to enhance code consistency and reliability. An iterative development methodology was adopted to allow rapid adjustments based on user needs. The core features of the system include user registration and login, order creation, FSM-based production status tracking, integration with shipment tracking (airwaybill), and customer data management. The implementation results indicate that the system successfully performs all primary functions and provides a more structured, efficient, and easily monitored ordering experience for both customers and administrators. Therefore, the system is expected to improve service professionalism and support data-driven decision-making in the creative textile industry.*

**Keywords:** *Finite State Machine, Node Js, TypeScript, Textile Print Ordering, Web Information System*

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang Masalah**

Mochi Studio merupakan studio yang bergerak dalam bidang design printing kain. Kurang lebih Mochi Studio sudah berjalan dari tahun 2017. Berkembangnya informasi teknologi mempengaruhi pendekatan penjualan Mochi Studio pada Pembeli. Dimana mulai banyaknya order yang berantakan karena masih manual, progress yang kurang ter rekap, dan tracking pengiriman kain yang sudah sesuai. Tiap-tiap proses dalam bisnis Mochi Studio ini punya permasalahan yang harus diselesaikan satu persatu.

Dalam jurnal "*Collaborative Systems – Finite State Machines*" oleh Ion Ivan, Cristian Ciurea, dan Sorin Pavel (2010), Finite State Machine (FSM) dijelaskan sebagai pendekatan formal untuk memodelkan perilaku sistem kolaboratif, khususnya dalam konteks sistem perbankan. FSM digunakan untuk memvisualisasikan transisi status dalam sistem, dimana setiap status merepresentasikan kondisi sistem tertentu, dan perubahan status ditentukan oleh masukan serta perintah yang diterima sistem.

Penelitian oleh Rifqi Aji Pratama *et al.* [1] merancang sistem informasi pemesanan berbasis *web* untuk layanan digital printing Medina Printing. Sistem ini bertujuan meningkatkan efisiensi layanan melalui fitur pemesanan online, pembayaran digital, dan pengunggahan desain. Sistem dikembangkan menggunakan metode *Rapid Application Development* (RAD), yang menekankan kecepatan pengembangan melalui iterasi dari tahap analisis hingga pengujian.

Sementara itu, Agustin *et al.* [2] menggunakan metode *Finite State Machine* (FSM) untuk mengatur alur logika dan kontrol perilaku dalam permainan tradisional Setatak berbasis

Android. FSM dipilih karena mampu merepresentasikan *state*, *event*, dan *action* selama interaksi pengguna, sehingga transisi antar status dalam game dapat berjalan sistematis dan terkendali.

Konsep ini sangat relevan dan dapat diterapkan secara langsung dalam sistem *e-commerce* Mochi Studio. Dalam sistem *e-commerce*, pengguna juga mengalami serangkaian transisi status mulai dari:

1. **State awal:** pengguna belum login
2. **State login:** pengguna berhasil masuk
3. **State browsing:** melihat dan memilih produk
4. **State checkout:** memasukkan alamat dan metode pembayaran
5. **State pembayaran:** proses validasi pembayaran
6. **State pesanan selesai/dikirim:** transaksi berhasil

FSM membantu memodelkan dan mengontrol alur proses tersebut secara terstruktur, meminimalkan *error*, dan memudahkan *debugging* serta pengembangan fitur lanjutan. Setiap status dalam sistem *e-commerce* didefinisikan dengan jelas, begitu juga dengan transisi dan tindakan yang terjadi pada setiap status. Ini membuat sistem menjadi lebih stabil, dapat diprediksi, dan mudah dikembangkan.

Penelitian ini membangun diagram status untuk memetakan transisi antar status dalam sistem kolaboratif menggunakan FSM, termasuk pengelolaan dokumen dan perintah. Dengan menggunakan aplikasi *Collaborative Multicash Servicedesk* sebagai studi kasus, dilakukan analisis kompleksitas sistem dan statistik seperti jumlah permintaan per kategori serta beban kerja agen.

Maka dengan ini Mochi Studio membutuhkan sistem *platform* yang dapat memfasilitasi pemesanan order yang punya *workflow* yang lebih profesional. Salah satu solusi yang dapat mengoptimalkan kinerja transaksional di Mochi Studio adalah dengan membuat “**Sistem Pemesanan Print Textile Dengan Model *Finite State Machine***”.

## **1.2. Identifikasi Permasalahan**

Adapun masalah-masalah yang menjadi fokus penyelesaian dalam program ini adalah:

1. Masalah Pemesanan yang berantakan karena masih manual dengan WhatsApp
2. Masalah *Workflow* atau pencatatan proses dari tiap orderan, dimana bisa satu dengan lainnya berbeda dan tidak berurutan
3. Masalah penerapan *airwaybill* pada *finishing* proses agar pembeli tahu bahwa pesanan mereka berjalan lancar
4. Manajemen pembeli yang lebih terstruktur sehingga bisa menerapkan model promosi yang sesuai karena masih manual

## **1.3. Perumusan Masalah**

Perlunya menerapkan *platform* manajemen transaksi dengan menggunakan model FSM agar dapat mengatasi masalah proses pemesanan

## **1.4. Maksud dan Tujuan**

### **1.4.1. Maksud**

Tugas Akhir ini bertujuan untuk merancang dan membangun Sistem Pemesanan Print Tekstil dengan Model *Finite State Machine* yang dapat mengotomatisasi dan mengoptimalkan proses pemesanan di Mochi Studio agar lebih efisien, terstruktur, dan profesional.

### **1.4.2. Tujuan**

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Membuat sistem pemesanan digital yang menggantikan metode manual via WhatsApp sehingga lebih terorganisir.
2. Menerapkan *workflow* berbasis *Finite State Machine* untuk memastikan setiap pesanan mengikuti alur proses yang sistematis dan tidak berantakan.
3. Mengembangkan sistem pencatatan status pemesanan agar setiap tahapan dalam produksi dapat dipantau secara *real-time*.
4. Mengintegrasikan fitur pelacakan pengiriman (*airwaybill tracking*) agar pelanggan dapat mengetahui status pesanan mereka dengan mudah.
5. Membangun sistem manajemen pelanggan yang lebih terstruktur sehingga dapat diterapkan strategi promosi berbasis data.

## **1.5. Metode Penelitian**

### **1.5.1. Teknik Pengumpulan Data**

#### **1. Observasi**

Observasi dilakukan dengan mengamati langsung proses bisnis di Mochi Studio, termasuk bagaimana pemesanan dilakukan secara manual, alur *workflow* yang tidak beraturan, serta kendala dalam pencatatan dan tracking pesanan. Hasil observasi ini menjadi dasar untuk memahami kebutuhan sistem yang lebih efisien.

#### **2. Wawancara**

Wawancara dilakukan dengan pihak-pihak yang terlibat dalam operasional Mochi Studio, seperti pemilik, staf, dan pelanggan. Wawancara ini bertujuan untuk menggali informasi mengenai:

- a. Kendala yang dialami dalam pemrosesan pesanan secara manual.
- b. Fitur apa saja yang dibutuhkan dalam sistem pemesanan agar lebih efektif.



- c. Harapan pelanggan terkait sistem pelacakan pesanan (*airwaybill tracking*).

### 3. Studi Pustaka

Studi pustaka dilakukan dengan mencari referensi dari buku, jurnal, atau penelitian sebelumnya terkait Finite State Machine dalam sistem pemesanan, manajemen pelanggan, dan integrasi sistem tracking pengiriman. Selain itu, studi pustaka juga digunakan untuk memahami standar pengembangan perangkat lunak yang sesuai dengan kebutuhan sistem.

#### 1.5.2. Model Pengembangan Sistem

Model pengembangan sistem akan menggunakan metode *Waterfall*. Dengan rincian sebagai berikut :

##### 1. Analisis Kebutuhan Software

Dari hasil observasi dan wawancara, dilakukan analisis kebutuhan perangkat lunak yang mencakup:

- a. Kebutuhan Fungsional: Sistem harus mampu menangani pemesanan, *workflow* order, tracking pengiriman, dan manajemen pelanggan.
- b. Kebutuhan Non-Fungsional: Sistem harus mudah digunakan, memiliki *interface* yang intuitif, serta dapat diakses oleh staf dan pelanggan.

##### 2. Desain

Berdasarkan analisis kebutuhan, dilakukan perancangan sistem yang mencakup:

- a. *Database Design*: Merancang struktur database untuk menyimpan data pemesanan, pelanggan, dan tracking pengiriman yang relevan menggunakan [Draw.io](https://draw.io) yang nantinya akan diimplementasikan pada *Database* jenis MySQL

- b. *Software Architecture*: Menggunakan arsitektur DDD dengan berbasis web yang sesuai dengan kebutuhan Mochi Studio serta menggunakan metode *waterfall* dalam pengembangannya.
- c. *User Interface (UI/UX) Design*: Mendesain antarmuka dengan *Prototype* agar mempermudah pengguna dalam melakukan pemesanan dan pelacakan pesanan.

### 3. *Code Generation*

Dalam tahap ini, implementasi sistem dilakukan menggunakan metode arsitektur *Waterfall* melalui pendekatan kebutuhan berdasarkan analisa laporan transaksi, Kemudian akan diimplementasikan pada program dengan teknologi pengembangan sebagai berikut :

- a. *Backend* menggunakan Node.js dengan Express.
- b. *Frontend* menggunakan Nuxt.js.
- c. *Database* menggunakan MySQL untuk menyimpan data transaksi dan pelanggan.

Sistem dikembangkan dengan pendekatan pemrograman berbasis objek (OOP) agar lebih modular dan mudah dikembangkan di masa depan.

### 4. *Testing*

Sistem diuji menggunakan *Black Box Testing*, yaitu:

- a. Menguji apakah fitur pemesanan berfungsi sesuai alur yang diharapkan.
- b. Memeriksa apakah status order berpindah sesuai dengan *Finite State Machine*.
- c. Menguji apakah *tracking* pengiriman dapat memberikan informasi secara *real-time* kepada pelanggan.

## 5. *Support*

Sistem yang dikembangkan perlu didukung dengan strategi pemeliharaan dan pengembangan di masa depan, seperti:

- a. Penyesuaian fitur berdasarkan umpan balik pengguna.
- b. Pengembangan fitur tambahan, misalnya sistem pembayaran otomatis atau integrasi dengan layanan ekspedisi.
- c. Peningkatan keamanan dan performa sistem untuk mengantisipasi pertumbuhan bisnis Mochi Studio.

### 1.6. Ruang Lingkup

Ruang lingkup pembuatan sistem ini adalah :

#### 1. Pengembangan Sistem Pemesanan

Penelitian ini berfokus pada sistem yang dibangun untuk memenuhi kebutuhan Mochi Studio dalam proses pemesanan, transaksional serta pengolahan pesanan.

#### 2. Fitur Aplikasi Dengan *Finite State Machine*

Aplikasi ini berfokus pada penerapan *Finite State Machine* pada sistem pemesanan sehingga aplikasi yang diterapkan dapat relevan serta mencukupi kebutuhan sistem yang diperlukan oleh Mochi Studio.

## 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 Dicetak", 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.

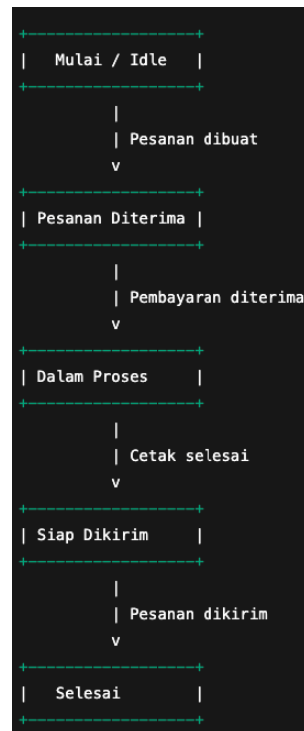
FSM memiliki implementasi yang lebih luas dalam berbagai sistem informasi, antara lain:

- 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].

## **BAB III**

### **ANALISA SISTEM BERJALAN**

#### **3.1. Tinjauan Institusi/Perusahaan**

Saat ini, Mochi Studio mengelola seluruh operasionalnya secara konvensional, dengan fokus utama pada penjualan dan pencatatan pesanan melalui *platform* media sosial seperti WhatsApp dan Facebook. Proses ini sangat bergantung pada interaksi manual, mulai dari penerimaan order hingga pencatatan detail pesanan. Setiap transaksi dicatat secara individual, yang berpotensi menimbulkan kendala dalam efisiensi dan akurasi data seiring dengan meningkatnya volume pesanan.

##### **3.1.1 Sejarah Institusi/Perusahaan**

Mochi Studio berdiri pada tahun 2016 yang awalnya sebagai tim *performance Cosplay Project* di daerah Tegal. Kemudian berubah menjadi usaha umkm penerima jasa pembuatan properti panggung dari mulai replika senjata, baju zirah dan juga baju panggung yang bisa disesuaikan dengan keinginan konsumen. Lebih jauh lagi pada tahun 2019 studio sempat memenangkan kompetisi World Cosplay Summit Regional dan berhasil menjadi finalis di Jakarta sehingga pesanan properti semakin banyak.

Dengan naiknya nama Mochi Studio di tengah tengah persaingan bisnis pembuatan properti yang ada di Indonesia menunjukkan bahwa bisnis digital didasari pada kepercayaan termasuk yang menjalankannya harus punya bukti bahwa dia profesional dan berbakat di bidangnya, contohnya dengan memenangkan perlombaan di berbagai kompetisi.

##### **3.1.2. Struktur Organisasi dan Fungsi**

Untuk struktur organisasi Mochi Studio tidaklah formal dan masih mengandalkan mitra serta internal tim. Siapa yang bekerja akan mendapatkan hasilnya begitu juga yang tidak. Dalam satu transaksi tidak semua tim terlibat, terkadang hanya beberapa saja. Hal inilah yang membuat Mochi Studio membutuhkan sistem

digitalisasi yang lebih terpantau dan menitik beratkan selesainya pesanan dengan segera secara profesional.

### **3.2. Proses Bisnis**

Dalam menjalankan bisnisnya selama ini Mochi Studio menggunakan langkah-langkah konvensional untuk berinteraksi dengan pelanggan. Pelanggan biasanya menemukan Mochi Studio dalam grup jual beli di sosial media kemudian memesan print tekstil atau properti melalui pesan.

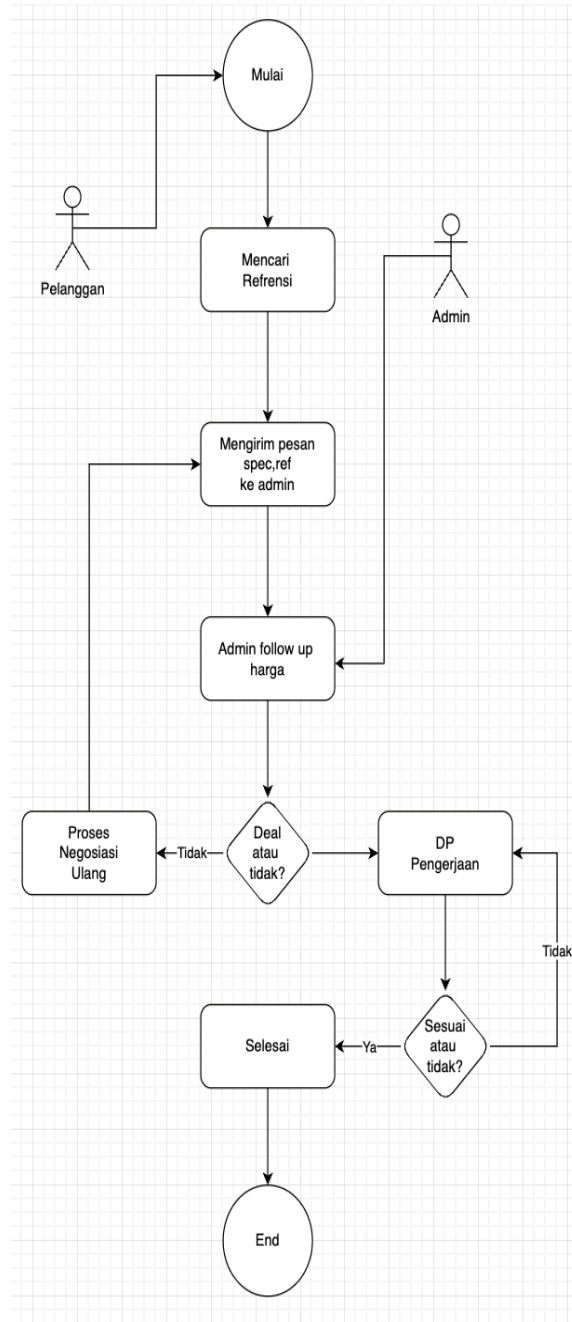
Secara ringkas proses bisnis yang diterapkan :

1. Pelanggan membuat sebuah pesan dan menanyakan detail dari pengerjaan.
2. Admin akan membalas dengan memberi estimasi, harga serta spesifikasi pada tiap harga, dalam proses ini sangat rumit dan seringkali terjadi *feedback* yang berulang karena sering mencocokkan dengan keadaan, fungsi properti ataupun budget dari pelanggan itu sendiri.
3. Kemudian ketika pelanggan membayar uang muka maka proses dikerjakan
4. Akan ada beberapa tahap seperti memfoto produk kemudian dikonfirmasi terlebih dahulu dan memastikan bahwa hasil jadi sudah sesuai ekspektasi pelanggan, jika tidak proses kembali ke pengerjaan revisi.
5. Jika telah memenuhi ekspektasi pelanggan maka pelanggan melakukan pembayaran penuh dan barang akan segera dikirim.

Secara teknis masih banyak proses yang berulang dan tidak efisien sehingga menimbulkan beberapa masalah, antara lain :

1. Pelacakan status dan proses pesanan yang saling bercampur sehingga prioritas pengerjaan menjadi terganggu
2. Acaknya sistem manual sehingga antrian bisa membuat manajemen diragukan profesionalismenya.

3. Akurasi laporan dari awal proses sampai akhir menjadi tidak terdefinisi dengan banyaknya orderan yang saling tumpang tindih



**Gambar 3. 1 Proses Bisnis**

### **3.3. Spesifikasi Dokumen Sistem Berjalan**

Spesifikasi dokumen sistem yang berjalan dan di implementasikan pada Mochi Studio hanya berupa percakapan daring dengan pelanggan menggunakan Messenger ataupun WhatsApp. Bisa disimpulkan dokumen data yang digunakan :

1. Catatan Pesanan Pelanggan
2. Riwayat percakapan sebagai akad kesepakatan
3. Harga yang disetujui
4. Referensi Design
5. Waktu linimasa yang disepakati dalam proses pengerjaan
6. Struk pembayaran

## BAB IV

### RANCANGAN SISTEM DAN PROGRAM USULAN

#### 4.1. Analisa Kebutuhan Software

Sistem pemesanan print tekstil merupakan aplikasi berbasis *web* yang dikembangkan untuk mendukung digitalisasi proses bisnis layanan cetak dan pemesanan properti secara terintegrasi. Aplikasi ini hadir sebagai alternatif dari cara pencatatan manual yang kerap menimbulkan risiko kesalahan data, kurang efisien, dan menyulitkan dalam pemantauan riwayat pemesanan maupun status produksi.

Melalui implementasi sistem ini, seluruh tahapan operasional mulai dari pendaftaran pelanggan, pemilihan jenis kain dan motif, penjadwalan, penghitungan total biaya, hingga pemantauan progres pesanan dapat dikelola secara *real-time*. Petugas dapat melakukan pencatatan transaksi dan memperbarui status pesanan, sedangkan administrator memiliki kewenangan lebih luas untuk memantau aktivitas sistem, mengelola data layanan cetak, pengguna, serta menyusun laporan pemesanan.

Pada sisi *backend*, sistem ini dibangun menggunakan *Express.js*, yaitu kerangka kerja minimalis dan fleksibel berbasis *Node.js* yang mendukung pengembangan RESTful API dengan performa tinggi. *Express* dipilih karena kemudahan dalam pengelolaan *routing*, *middleware*, serta integrasi ke berbagai basis data modern. Sementara pada sisi *frontend*, aplikasi memanfaatkan *Nuxt.js*, framework *Vue.js* yang mendukung rendering sisi server (*server-side rendering*) sehingga menghasilkan antarmuka yang cepat dan optimal di berbagai perangkat.

Untuk mengatur alur proses pemesanan dan transisi status secara terstruktur, diterapkan pendekatan *Finite State Machine* (FSM). Dengan model ini, setiap pesanan akan melalui sejumlah keadaan (*state*) yang terdefinisi, misalnya Pesanan Diterima, Proses Cetak, Selesai, dan Diambil, sehingga pergerakan data lebih terkendali dan transparan.



Aplikasi ini dapat diakses melalui berbagai peramban *web* seperti Google Chrome, Mozilla Firefox, atau lainnya. Antarmuka pengguna dirancang agar intuitif, responsif, dan mudah digunakan oleh operator toko dengan latar belakang teknis yang beragam.

#### **4.1.1. Kebutuhan Fungsional Sistem:**

- a. **Pencatatan dan Pemantauan Transaksi:** Sistem harus mampu mencatat dan memantau setiap tahapan pesanan transaksi, mulai dari pemesanan awal, konfirmasi pembayaran, proses produksi, hingga pengiriman atau pengambilan. Ini mencakup riwayat transaksi lengkap untuk memudahkan pemantauan dan audit.
- b. **Manajemen Data Produk dan Harga:** Mengelola katalog produk transaksi (jenis kain, teknik cetak, ukuran, dll.) dan struktur harga yang fleksibel, termasuk diskon atau promosi. Informasi harga harus selalu terkini dan transparan.
- c. **Dashboard Pemantauan:** Menyediakan *dashboard* interaktif yang menampilkan statistik pesanan, status produksi, ketersediaan bahan baku (jika relevan), dan insight penjualan secara *real-time* untuk membantu manajemen dan staf dalam pengambilan keputusan.
- d. **Pengelolaan Data Pengguna:** Fitur manajemen pengguna yang komprehensif untuk mengelola data pelanggan dan admin, dengan hak akses yang berbeda sesuai peran masing-masing.
- e. **Laporan dan Dokumentasi:** Menghasilkan laporan transaksi, pesanan, dan dokumen terkait lainnya secara otomatis untuk kebutuhan administrasi, pelaporan keuangan, dan analisis bisnis.
- f. **Fitur Pemesanan Online:** Memungkinkan pelanggan untuk melakukan pemesanan transaksi secara online melalui *website*, memilih spesifikasi produk, mengunggah desain, dan menyelesaikan proses *checkout* dengan mudah.

#### 4.1.2. Kebutuhan Non-Fungsional

Untuk Sistem Pemesanan transaksi Mochi Studio, kebutuhan non-fungsional sangat krusial untuk memastikan pengalaman pengguna yang optimal dan operasional yang efisien:

- a. **Aksesibilitas Berbasis Website:** Sistem ini akan dikembangkan sebagai aplikasi *web*, memungkinkan aksesibilitas kapan saja dan di mana saja melalui perangkat apa pun yang terhubung internet. Ini menjamin fleksibilitas operasional bagi Mochi Studio dan kemudahan bagi pelanggan.
- b. **Keamanan Data dan Transaksi:** Keamanan adalah prioritas utama. Sistem akan menerapkan teknologi autentikasi yang kuat untuk melindungi data pelanggan, detail pesanan, dan transaksi. Ini termasuk enkripsi data, otorisasi peran pengguna (admin vs. pelanggan), dan perlindungan terhadap akses tidak sah untuk mencegah penyalahgunaan.
- c. **Kemudahan Penggunaan (*User-Friendly*):** Antarmuka pengguna (UI) harus intuitif dan responsif, dirancang agar mudah digunakan oleh staf Mochi Studio maupun pelanggan tanpa memerlukan pelatihan teknis yang rumit. Tujuannya adalah meminimalkan hambatan adaptasi dan memaksimalkan efisiensi.
- d. **Skalabilitas:** Sistem harus dirancang untuk dapat berkembang seiring dengan pertumbuhan Mochi Studio. Ini berarti sistem mampu menangani peningkatan volume pesanan, jumlah pelanggan, dan penambahan fitur baru di masa mendatang tanpa mengorbankan kinerja.
- e. ***Responsive Design*** : Aplikasi dirancang dengan kemampuan responsif di berbagai ukuran layar sehingga membuat pelanggan mudah beradaptasi dalam aplikasi baik dengan gawai yang kecil maupun ukuran besar.
- f. ***Database*** : Aplikasi menyimpan *database* dengan keamanan tinggi menggunakan jenis *database* MySQL

## 4.2. Desain

### 4.2.1 Desain Pemodelan Sistem

Sistem Pemesanan transaksi untuk Mochi Studio berbasis *website* dengan teknologi Node.js akan dirancang dengan pendekatan modern, mengadopsi praktik terbaik dalam pengembangan aplikasi *web* dan disesuaikan dengan alur bisnis transaksi.

#### 1. Model Konseptual dan Arsitektur Sistem

- a. **Arsitektur Berbasis *Web* dengan Node.js sebagai *Backend*:** Node.js akan menjadi fondasi *backend*, menangani logika bisnis, API, dan pengelolaan database secara efisien dan *scalable*. Ini ideal untuk mendukung pemrosesan pesanan *real-time* dan pelacakan status melalui *Finite State Machine*.
- b. ***Frontend* Berbasis Web Responsif:** Antarmuka pengguna akan dibangun menggunakan *framework* JavaScript modern NuxtJs. Ini akan memastikan sistem mudah diakses dari berbagai perangkat (desktop, tablet, smartphone) dengan tampilan yang adaptif dan pengalaman pengguna yang lancar.
- c. ***Database Relasional*:** MySQL atau PostgreSQL akan digunakan untuk menyimpan data secara terstruktur dan terintegrasi. Ini mencakup data pengguna (pelanggan dan admin), katalog produk (jenis kain, metode cetak, ukuran, dll.), detail pesanan, status transaksi, dan riwayat produksi. Struktur database akan dirancang dengan *Physical Data Model* yang valid untuk memastikan integritas dan efisiensi *query*.
- d. **RESTful API:** Komunikasi antara *frontend* dan *backend* akan diatur melalui RESTful API. Ini memungkinkan pertukaran data yang efisien dan modular, memudahkan pengembangan dan integrasi di masa depan.
- e. **Keamanan Autentikasi dan Otorisasi:** Implementasi JSON Web Tokens (JWT) atau mekanisme autentikasi serupa akan digunakan untuk mengelola akses pengguna. Ini memastikan bahwa hanya

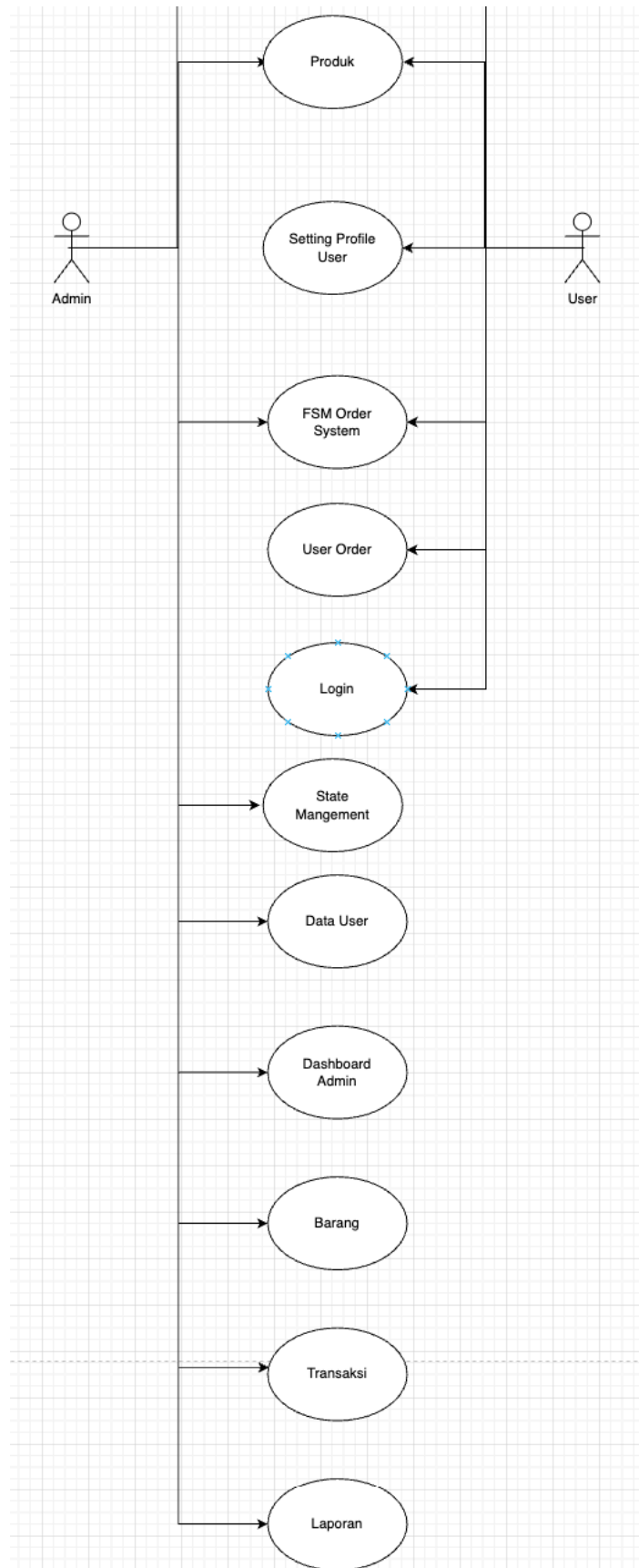
pengguna terdaftar (pelanggan) dan admin yang berwenang yang dapat mengakses fitur-fitur yang relevan.

## **2. Diagram Pemodelan Sistem (UML)**

Desain sistem akan divisualisasikan menggunakan diagram UML (*Unified Modeling Language*) untuk memberikan gambaran yang jelas tentang fungsionalitas dan struktur:

### **a. *Use Case Diagram*:**

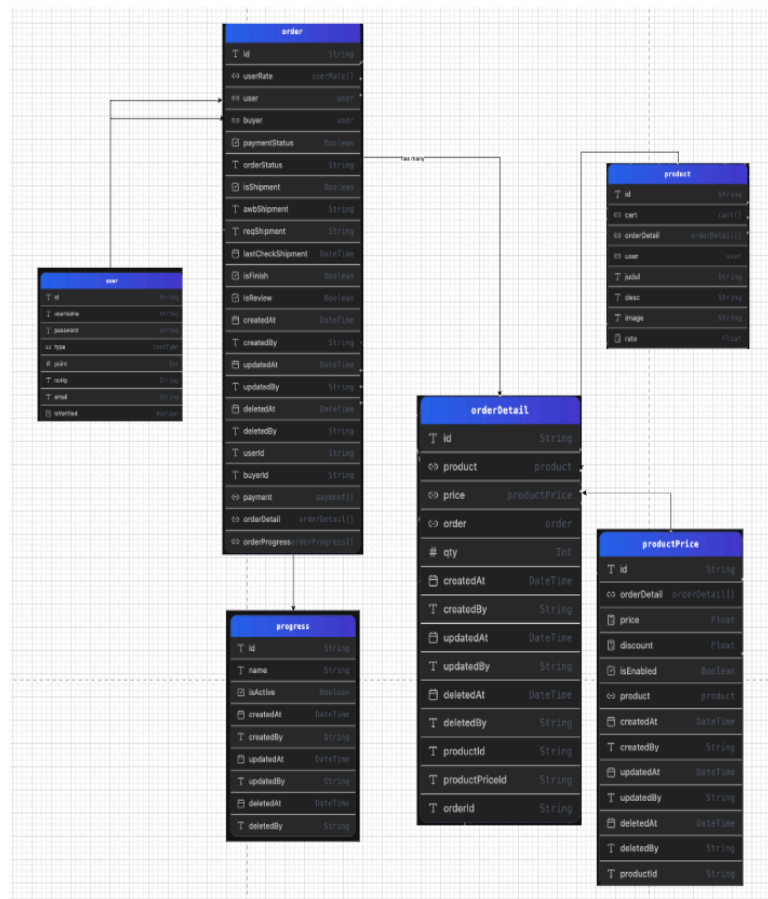
*Use Case Diagram* adalah gambaran interaksi dari pengguna sistem terhadap fitur yang tersedia. Dalam hal ini sistem yang dibangun mempunyai fungsi: Login, Manajemen Profil Pengguna, Pemesanan Barang (termasuk unggah desain, pemilihan spesifikasi), Pelacakan Status Pesanan, Konfirmasi Pembayaran, Pengelolaan Data Produk, Manajemen Pesanan, Pembaruan Status Produksi, dan Pelaporan.



**Gambar 4. 1 Use Case Diagram**

### b. Class Diagram

*Class Diagram* adalah alat untuk membantu memvisualisasikan atau menggambarkan struktur kelas yang terdapat pada suatu sistem program. *Class* menggambarkan kelompok suatu objek bersama properti, operasi dan relasi yang sama[8].



**Gambar 4. 2 Class Diagram**

## 3. Komponen dan Modul

Sistem Pemesanan transaksi Mochi Studio akan memiliki beberapa modul utama yang bekerja sama untuk mempermudah seluruh proses, baik bagi pelanggan maupun tim internal.

- a. **Manajemen Pengguna.** Modul ini memungkinkan pelanggan untuk mendaftar, masuk ke akun mereka, dan mengelola informasi profil pribadi. Admin juga bisa mengatur profil mereka sendiri. Yang

terpenting, modul ini memastikan setiap orang hanya memiliki akses ke fitur yang sesuai dengan peran mereka; pelanggan dapat memesan dan melihat riwayat pesanan, sementara admin bisa mengelola semua data dan transaksi.

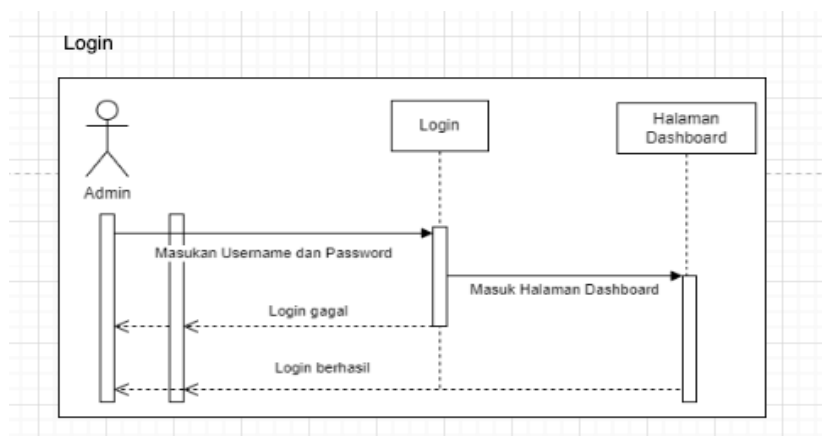
- b. **Manajemen Produk** akan menangani semua informasi tentang layanan transaksi Mochi Studio. Ini termasuk memasukkan dan memperbarui detail seperti jenis kain yang tersedia, teknik cetak, pilihan ukuran, dan struktur harga yang fleksibel. Modul ini memastikan Mochi Studio dapat dengan mudah menambahkan atau mengubah spesifikasi produk sesuai kebutuhan pasar.
- c. **Pemesanan.** Di sini, pelanggan dapat melakukan pemesanan secara online, mengunggah desain mereka, memilih spesifikasi produk yang diinginkan, dan melihat perkiraan biaya. Sistem secara otomatis akan memvalidasi kelengkapan pesanan sebelum diserahkan.
- d. **Pelacakan Pesanan (FSM).** Modul ini memungkinkan pelacakan status pesanan secara real-time menggunakan konsep *Finite State Machine*. Bayangkan status pesanan bergerak dari "Menunggu Pembayaran", lalu ke "Desain Diverifikasi", kemudian "Dalam Produksi", hingga "Siap Kirim/Ambil", dan akhirnya "Selesai". Admin akan memiliki kemampuan untuk memperbarui status ini sesuai dengan kemajuan pesanan.
- e. **Pengelolaan Transaksi.** Modul ini memungkinkan admin untuk mengelola semua aspek pesanan, mulai dari mengonfirmasi pembayaran, memperbarui status produksi, hingga menangani masalah yang mungkin timbul. Admin juga bisa memberikan otorisasi atau bahkan membatalkan pesanan jika diperlukan.

Untuk membantu pengambilan keputusan, ada modul Laporan & Statistik. Ini akan menghasilkan berbagai laporan digital, seperti laporan penjualan, daftar pesanan, dan statistik produksi. Data ini sangat berharga bagi Mochi Studio untuk menganalisis kinerja bisnis mereka.

#### 4. Sequence Diagram

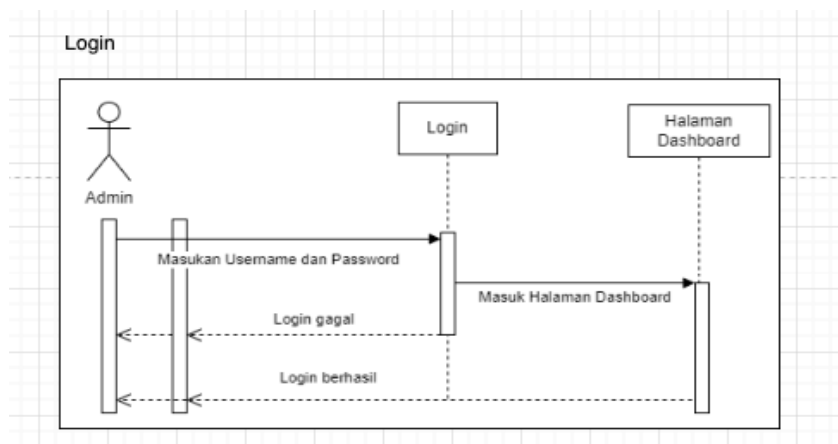
*Sequence Diagram* digunakan untuk menggambarkan alur komunikasi antar objek pada satu use case tertentu secara berurutan. Diagram ini menunjukkan bagaimana alur kerja dari *login* dimulai dari user hingga sistem melakukan autentikasi dan memberikan respon, baik sukses maupun gagal.

##### a. Sequence Diagram Pendaftaran



**Gambar 4. 3 Sequence Diagram Pendaftaran**

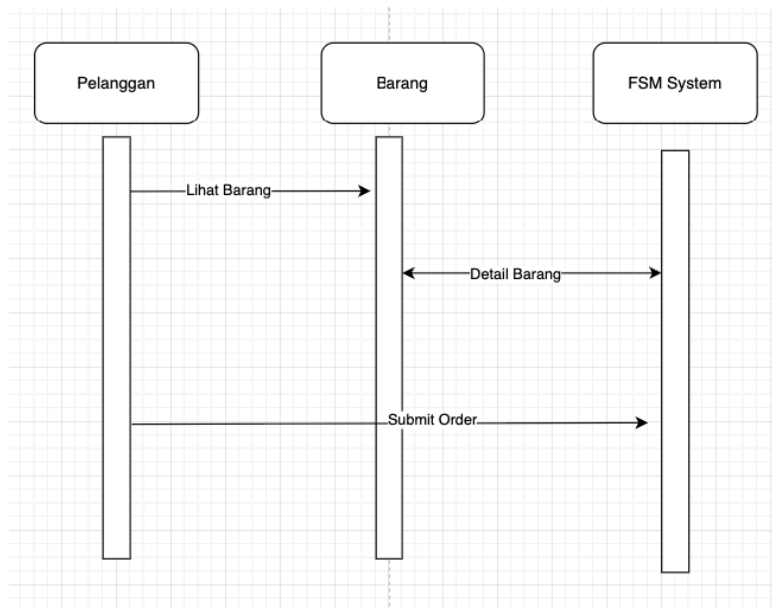
##### b. Sequence Diagram Login



**Gambar 4. 4 Sequence Diagram Login**

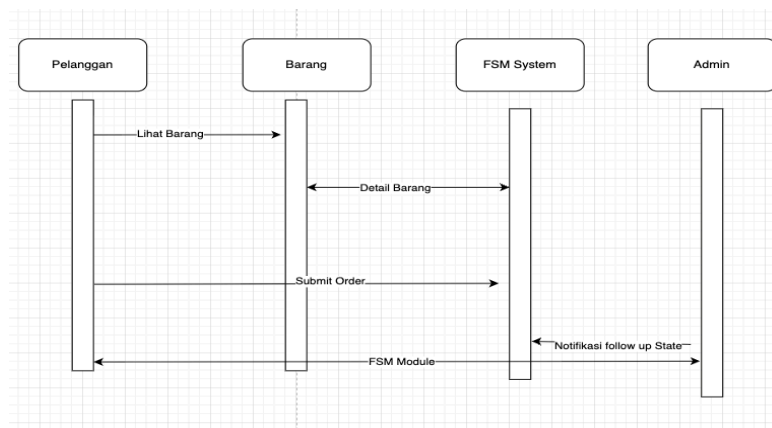


c. *Sequence Diagram Checkout*



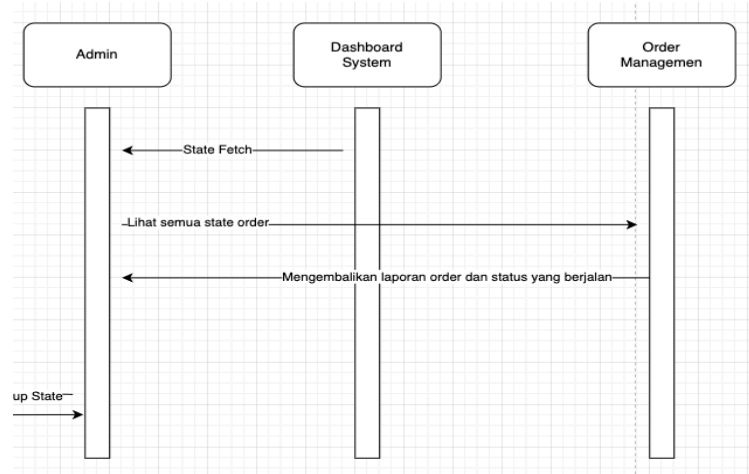
**Gambar 4. 5 Sequence Diagram Checkout**

d. *Sequence Diagram Pemesanan dengan FSM Model*



**Gambar 4. 6 Sequence Diagram Pemesanan**

e. *Sequence Diagram Laporan dan Pelacakan Pesanan*



**Gambar 4. 7 Sequence Diagram Laporan Pesanan**

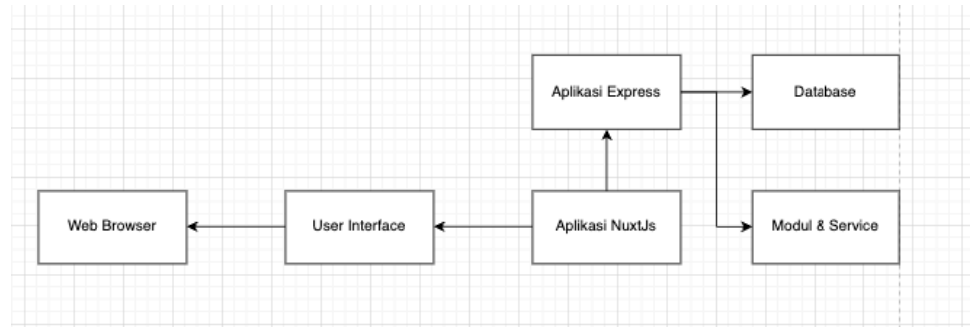
## 5. Teknologi Pendukung

Untuk memastikan kinerja optimal dan skalabilitas, sistem ini akan dibangun dengan teknologi modern dan terbukti:

- Backend:** Node.js dengan *framework* Express.js untuk membangun API yang efisien dan scalable dalam menangani logika bisnis dan interaksi database.
- Database:** MySQL atau PostgreSQL akan digunakan sebagai *database* relasional untuk menyimpan data pengguna, produk, pesanan, dan status transaksi secara terstruktur.
- Frontend:** Nuxt.js (berbasis Vue.js) akan digunakan untuk membangun antarmuka pengguna yang responsif dan performa tinggi, mendukung *Server-Side Rendering* (SSR) untuk SEO dan pengalaman pengguna yang lebih baik.
- Autentikasi:** JSON Web Tokens (JWT) akan diimplementasikan untuk mengelola sesi pengguna dan mengamankan akses ke berbagai modul sistem.
- Deployment:** Sistem akan diluncurkan pada layanan *cloud hosting* (misal: AWS, DigitalOcean, Heroku) untuk memastikan aksesibilitas luas dan ketersediaan tinggi dan bagian *front end* akan diluncurkan dengan layanan netlify.

## 6. Component Diagram

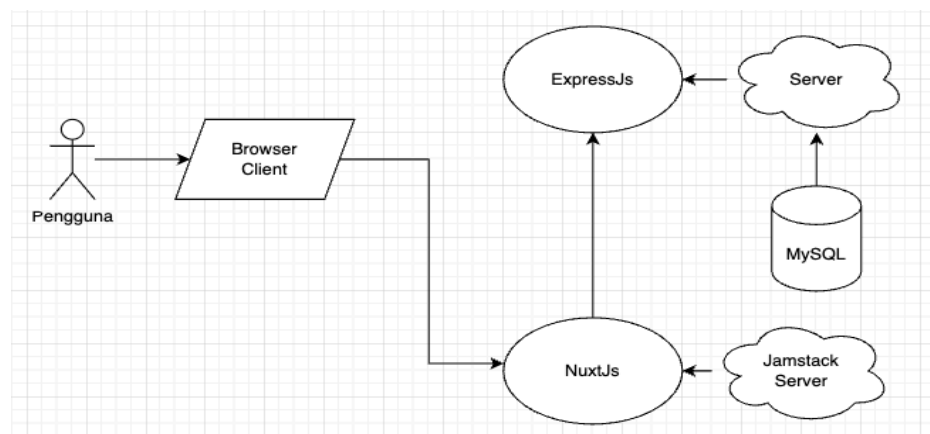
*Component* diagram menggambarkan bagaimana perangkat lunak berinteraksi untuk membangun sebuah sistem yang nantinya digunakan oleh pengguna. Berikut gambaran bagaimana sistem pemesanan ini dibangun :



**Gambar 4. 8 Component Diagram**

## 7. Deployment Diagram

Dalam *deployment* diagram digambarkan bagaimana nanti sistem yang telah dikembangkan dijalankan dalam konfigurasi fisik baik secara lokal maupun *live*. Dalam diagram ada beberapa node yang saling terhubung antara aplikasi pada sisi server dan aplikasi yang diakses pelanggan. Berikut gambaran dari deployment yang dilakukan oleh sistem :



**Gambar 4. 9 Deployment Diagram**

#### 4.2.2 Desain Pemodelan Data

Desain pemodelan data untuk aplikasi pemesanan produk kustom Mochi Studio berbasis website menggunakan Node.js secara umum melibatkan beberapa komponen utama. Komponen-komponen ini dirancang untuk merefleksikan proses bisnis inti Mochi Studio, meliputi manajemen produk kustom, data pelanggan, transaksi pemesanan, dan pengelolaan progres pengerjaan produk.

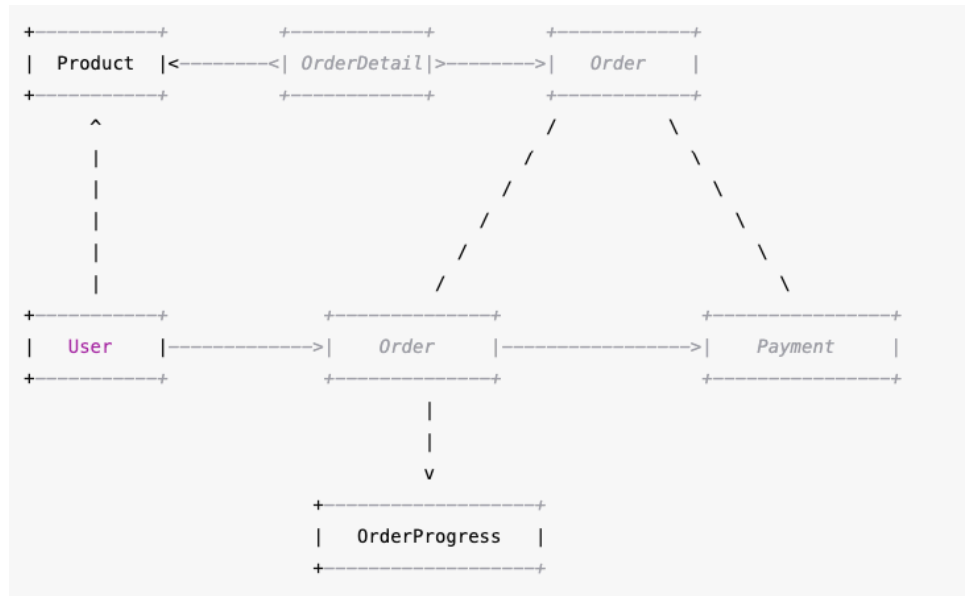
Desain ini mengadaptasi praktik terbaik dalam pengembangan sistem berbasis website, dengan penyesuaian pada teknologi Node.js sebagai backend modern dan penggunaan Prisma ORM untuk interaksi database yang efisien dan terstruktur.

##### 1. Entitas dan Relasi Utama

Berdasarkan skema Prisma yang saya gunakan untuk pengembangan sistem ini, berikut adalah entitas utama beserta relasinya:

- a. Product (1) — (N) OrderDetail: Satu jenis produk kustom Mochi Studio (misalnya, desain mug tertentu, pola kaos) dapat disertakan dalam banyak detail transaksi pemesanan. Relasi ini menunjukkan bahwa satu item produk master bisa dipesan berkali-kali oleh pelanggan yang berbeda.
- b. User (Pelanggan) (1) — (N) Order: Satu pelanggan dapat melakukan banyak transaksi pemesanan. Entitas User di sini berperan sebagai pembeli (*buyer*).
- c. Order (1) — (N) OrderDetail: Satu transaksi pemesanan (Order) dapat terdiri dari banyak item produk Mochi Studio yang berbeda (OrderDetail), misalnya satu pesanan bisa mencakup mug kustom dan kaos kustom.
- d. Order (1) — (N) Payment: Satu order dapat memiliki banyak *record* pembayaran. Ini mengakomodasi skenario seperti pembayaran uang muka (DP) dan pelunasan, atau riwayat upaya pembayaran yang gagal.

- e. Order (1) — (N) OrderProgress: Setiap order akan memiliki banyak *record progress* yang secara spesifik menunjukkan tahapan pengerjaan kustom yang telah atau sedang dijalani. Misalnya, untuk satu order mug kustom, akan ada *progress* "Desain Disetujui", "Proses Cetak Dimulai", dan seterusnya.



**Gambar 4. 10 Ilustrasi Logika *Entitas***

## 2. Teknologi dan Pendekatan

- Backend: Saya menggunakan Node.js dengan *framework* Express.js untuk membangun API. Ini memungkinkan sistem untuk berinteraksi secara efisien antara frontend dan database.
- Database: Saya memanfaatkan database relasional MySQL/PostgreSQL yang diakses melalui Prisma ORM. Prisma dipilih karena kemampuannya dalam mengelola skema database secara type-safe dan mempermudah operasi CRUD (Create, Read, Update, Delete).
- Desain Sistem: Saya menerapkan metode *Waterfall* untuk pendekatan pengembangan sistem. Pemodelan dilakukan dengan UML (Unified Modeling Language), mencakup Use Case Diagram untuk mendefinisikan fungsionalitas sistem dari sudut pandang pengguna, Class Diagram untuk memvisualisasikan struktur data

dan relasi antar entitas, serta *Sequence Diagram* untuk menggambarkan alur interaksi antarkomponen sistem secara jelas.

- d. Fitur Tambahan: Sistem ini juga akan dilengkapi dengan fitur notifikasi (misalnya melalui email atau WhatsApp *Gateway*) untuk memberitahukan status order atau progres pengerjaan kepada pelanggan.

### 3. Contoh Struktur Koleksi/Tabel (Prisma/SQL)

Berikut adalah representasi struktur tabel atau koleksi utama dalam database, disesuaikan dari skema Prisma yang saya gunakan untuk Mochi Studio:

```
// Model Product (Produk Mochi Studio)

{
  "id": "String (UUID)",
  "judul": "String",
  "desc": "String",
  "image": "String (URL gambar)",
  "qty": "Int (Stok)",
  "type": "String (Kategori produk, misal: 'Mug Custom', 'Kaos Custom')",
  "isPo": "Boolean (Indikator Pre-Order)",
  "process": "String (Deskripsi alur pengerjaan kustom)",
  "userId": "String (FK ke User - pemilik produk)",
  "createdAt": "DateTime",
  "updatedAt": "DateTime"
}
```

```

// Model User (Pelanggan/Admin)

{

    "id": "String (UUID)",

    "username": "String (Unique)",

    "email": "String (Unique)",

    "password": "String",

    "noHp": "String (Nomor Telepon)",

    "displayName": "String",

    "type": "Enum (UserType - User/Admin)",

    "createdAt": "DateTime",

    "updatedAt": "DateTime"

}


// Model Order (Pemesanan)

{

    "id": "String (UUID)",

    "userId": "String (FK ke User - pemilik produk/Mochi Studio)",

    "buyerId": "String (FK ke User - pelanggan)",

    "paymentStatus": "Boolean (True jika sudah dibayar penuh)",

    "orderStatus": "String (Dikendalikan FSM: PENDING, PAID, PROCESSING, SHIPPED, etc.)",

    "isShipment": "Boolean (Indikator siap kirim)",

    "awbShipment": "String (Nomor Resi Pengiriman)",

    "isFinish": "Boolean (Indikator order selesai)",

```

```

    "isReview": "Boolean (Indikator sudah di-review)",

    "createdAt": "DateTime",

    "updatedAt": "DateTime"

}

// Model OrderDetail (Detail Item Pemesanan)

{

    "id": "String (UUID)",

    "orderId": "String (FK ke Order)",

    "productId": "String (FK ke Product)",

    "productPriceId": "String (FK ke ProductPrice - harga spesifik yang
dipilih)",

    "qty": "Int (Jumlah produk)",

    "createdAt": "DateTime",

    "updatedAt": "DateTime"

}

// Model Progress (Master Data Tahapan Pengerjaan Kustom)

{

    "id": "String (UUID)",

    "name": "String (Nama tahapan, misal: 'Desain Disetujui', 'Produksi
Dimulai')",

    "isActive": "Boolean",

    "createdAt": "DateTime",

    "updatedAt": "DateTime"

```



```
}
```

```
// Model OrderProgress (Progres Pengerjaan Spesifik per Order)
```

```
{
```

```
  "id": "String (UUID)",
```

```
  "orderId": "String (FK ke Order)",
```

```
  "progressId": "String (FK ke Progress)",
```

```
  "status": "String (Status tahapan ini, misal: 'completed', 'pending')",
```

```
  "timestamp": "DateTime (Waktu update progres)",
```

```
  "createdAt": "DateTime",
```

```
  "updatedAt": "DateTime"
```

```
}
```

#### 4. Alur Sistem

Alur sistem pemesanan Mochi Studio berpusat pada pengelolaan status order yang dinamis, tanpa adanya pelacakan kiriman secara eksternal melainkan hanya pencatatan AWB.

- a. Melakukan Pemesanan: Pelanggan memilih produk Mochi Studio, termasuk opsi kustomisasi jika ada, dan menyelesaikan proses pemesanan melalui website. Sistem kemudian mencatat order baru dengan status awal (orderStatus) "Pending" dan detail produk (OrderDetail).
- b. Pembayaran: Setelah pemesanan, pelanggan melakukan pembayaran. Sistem akan memverifikasi pembayaran dan memperbarui paymentStatus pada order terkait. Jika pembayaran berhasil, orderStatus akan bertransisi ke "Paid".
- c. Proses Pengerjaan (Mochi Studio): Setelah order berstatus "Paid", tim Mochi Studio akan memulai proses pengerjaan

produk kustom. Setiap tahapan pengerjaan (misalnya, persetujuan desain, produksi, quality control) akan dicatat secara detail dalam OrderProgress, memberikan visibilitas kepada pelanggan mengenai kemajuan order mereka. orderStatus utama akan berada di "Processing" selama tahapan ini.

- d. Penyelesaian Pengerjaan & Siap Kirim: Setelah semua tahapan pengerjaan selesai, orderStatus akan bertransisi ke "Ready for Shipment".
- e. Pengiriman: Tim Mochi Studio akan menyiapkan produk untuk pengiriman. Nomor resi pengiriman (*airwaybill*) akan dicatat pada entitas Order, dan orderStatus akan diperbarui menjadi "Shipped".
- f. Konfirmasi Penerimaan: Setelah produk diterima oleh pelanggan, mereka dapat melakukan konfirmasi penerimaan di sistem. orderStatus kemudian akan berubah menjadi "Delivered".
- g. Penyelesaian Order: Setelah produk diterima dan mungkin review diberikan oleh pelanggan, orderStatus akan bertransisi menjadi "Completed", menandakan berakhirnya siklus order.
- h. Admin Mengelola Operasi: Admin Mochi Studio memiliki peran sentral dalam mengelola data produk (product), memantau dan memperbarui status setiap order (orderStatus), mengelola progres pengerjaan (OrderProgress), dan mengelola data pelanggan (User). Sistem juga akan menyediakan laporan penjualan yang relevan untuk analisis bisnis.

Desain ini memastikan setiap langkah dalam proses pemesanan Mochi Studio terekam dengan jelas, dari awal hingga akhir, dengan *feedback* yang relevan kepada pelanggan dan alat manajemen yang komprehensif untuk Mochi Studio.

#### 4.2.3 Desain *User Interface*

Dalam perancangan sistem pemesanan untuk Mochi Studio, saya mengutamakan desain *User Interface* (UI) yang mampu menyediakan kemudahan penggunaan, kejelasan informasi, dan responsivitas. Tujuan utamanya adalah agar pelanggan dapat melakukan pemesanan produk kustom Mochi Studio dengan cepat, nyaman, dan terpandu dengan baik. Mengingat karakteristik produk Mochi Studio yang memungkinkan personalisasi dan sistem pre-order, UI yang dirancang harus mampu memfasilitasi interaksi yang dinamis dan memberikan *feedback progress* yang transparan kepada pengguna.

##### 1. Pendekatan Desain

Saya mengadopsi dua pendekatan utama dalam perancangan UI sistem ini:

- a. **Metode *Design Thinking***: Saya menerapkan tahapan *Design Thinking* yang berorientasi pada pengguna, meliputi *Empathize*, *Define*, *Ideate*, *Prototype*, dan *Testing*. Pendekatan ini esensial untuk memastikan desain UI benar-benar memahami kebutuhan unik pelanggan Mochi Studio, terutama terkait proses personalisasi produk dan alur pre-order yang mungkin kompleks. Melalui tahapan ini, saya dapat mendefinisikan masalah pengguna secara akurat, menghasilkan ide-ide solusi yang kreatif, membangun prototipe untuk visualisasi, dan melakukan pengujian langsung dengan calon pengguna untuk memvalidasi efektivitas dan kemudahan penggunaan desain.
- b. ***Human-Centered Design (HCD)***: Fokus saya pada *Human-Centered Design* (HCD) bertujuan untuk menciptakan tampilan yang intuitif dan mempermudah pelanggan di setiap langkah proses pemesanan. Mulai dari pemilihan produk, konfigurasi kustomisasi, hingga pelacakan progres pengerjaan dan pengiriman, setiap interaksi dirancang agar seamless dan efisien mungkin bagi pengguna.

## 2. Komponen Utama UI

Berikut adalah deskripsi komponen UI utama yang saya rancang untuk *website* pemesanan Mochi Studio, disesuaikan dengan alur bisnis dan struktur data yang telah dibahas:

*Tabel IV. 1 Tabel Halaman Website*

Halaman / Komponen	Deskripsi Fungsionalitas	Contoh Fitur Utama
Halaman <i>Login/Registrasi</i>	Berfungsi sebagai gerbang autentikasi bagi pengguna (pelanggan) untuk mengakses fitur-fitur pemesanan.	Menyediakan <i>form input</i> <i>username/email</i> dan <i>password</i> , tombol <i>login</i> , opsi "Lupa Password", serta tautan untuk registrasi akun baru. Pada <i>form</i> registrasi, pengguna mengisi data pribadi seperti nama, email, nomor HP, dan <i>display name</i> .
Halaman Beranda	Halaman ini dirancang untuk menyajikan sorotan produk, promosi terkini, atau koleksi produk unggulan Mochi Studio.	Mencakup banner promosi, <i>carousel</i> produk-produk pilihan, serta navigasi kategori produk untuk memudahkan eksplorasi.

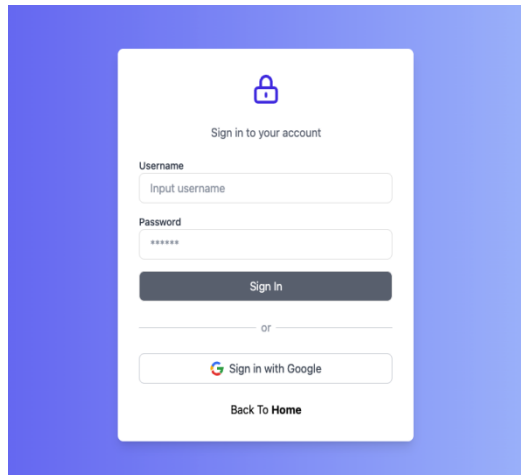
Halaman Produk & Detail	Halaman ini menampilkan daftar produk yang tersedia di Mochi Studio. Saat pengguna memilih produk, halaman detail akan menampilkan informasi yang lebih komprehensif, termasuk opsi kustomisasi atau spesifikasi pre-order.	Pada Daftar Produk, ditampilkan foto, judul, harga, dan tag relevan (isPo, best seller). Detail Produk mencakup galeri gambar, deskripsi ( <i>desc</i> ), rata-rata rating ( <i>rate</i> ), ulasan pelanggan ( <i>productComment</i> ), opsi kustomisasi (jika data produk mendukung), pilihan <i>productPrice</i> (harga dan diskon), indikator ketersediaan stok ( <i>qty</i> ), tombol "Tambah ke Keranjang", dan informasi <i>process</i> (alur kerja kustom) untuk produk personalisasi.
Halaman <i>Checkout</i>	Merupakan tahap final dalam proses pemesanan, di mana pengguna melengkapi data pengiriman, meninjau ringkasan order, dan memilih metode pembayaran.	Meliputi input alamat pengiriman ( <i>userAddress</i> ), pilihan jasa pengiriman, ringkasan <i>OrderDetail</i> yang mencakup <i>product</i> , <i>productPrice</i> , <i>qty</i> , informasi <i>paymentStatus</i> , dan tombol "Konfirmasi Pesanan".
Halaman Status Pesanan	Ini adalah komponen krusial yang menyediakan informasi terkini mengenai status setiap order pelanggan, termasuk progres pengerjaan dan pelacakan pengiriman. Halaman ini mengintegrasikan konsep <i>Finite State Machine</i> (FSM) yang telah saya rancang.	Daftar Order menampilkan <i>orderStatus</i> untuk setiap pesanan. Pada Detail Order, pengguna dapat melihat state FSM ( <i>orderStatus</i> seperti PENDING, PROCESSING, SHIPPED), detail <i>orderProgress</i> (misalnya "Desain Disetujui", "Produksi Dimulai"), nomor resi ( <i>awbShipment</i> ), <i>tracking</i> pengiriman, riwayat transaksi ( <i>payment</i> ), tombol konfirmasi penerimaan ( <i>confirmDelivery</i> ), serta opsi untuk memberikan <i>review</i> ( <i>isReview</i> ).

Admin Dashboard/Panel	Interface ini dirancang untuk manajemen internal Mochi Studio, mencakup pengelolaan produk, stok, transaksi, pengguna, serta pemantauan detail <i>progress order</i> .	Manajemen Produk (CRUD untuk product, productPrice, productGallery). Manajemen Order (melihat daftar order, mengubah orderStatus yang memicu FSM, update orderProgress— misalnya menandai "Desain Selesai", input awbShipment). Manajemen Pengguna (CRUD user). Laporan seperti statistik penjualan dan order berdasarkan status.
--------------------------	---	---

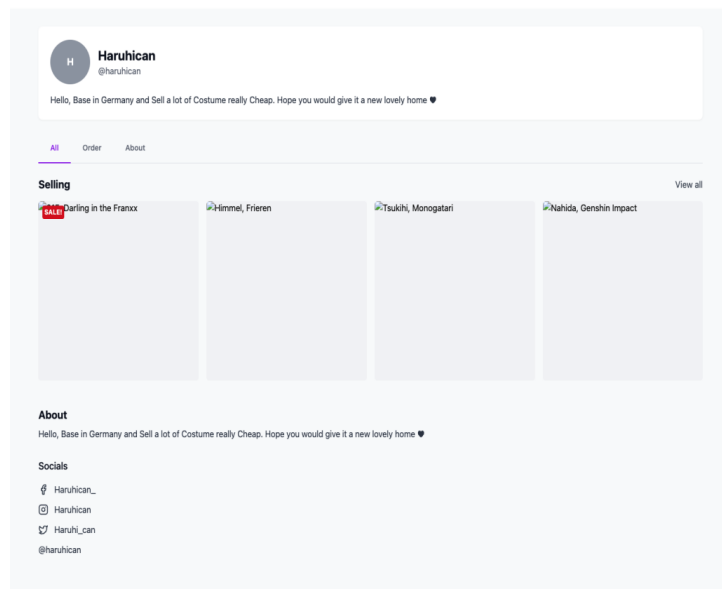
Berikut *User Interface* yang ada pada sistem :

The image shows a registration form titled "Validate your account" with a lock icon. The form includes three input fields: "Username" (placeholder: "Input username"), "Email" (placeholder: "Input email"), and "No Hp" (placeholder: "Input no hp without 0/82"). Below these fields is a checkbox labeled "I agree with the terms and conditions". At the bottom of the form is a blue "Register" button and a link that says "Back To Home". The entire form is centered on a blue gradient background.

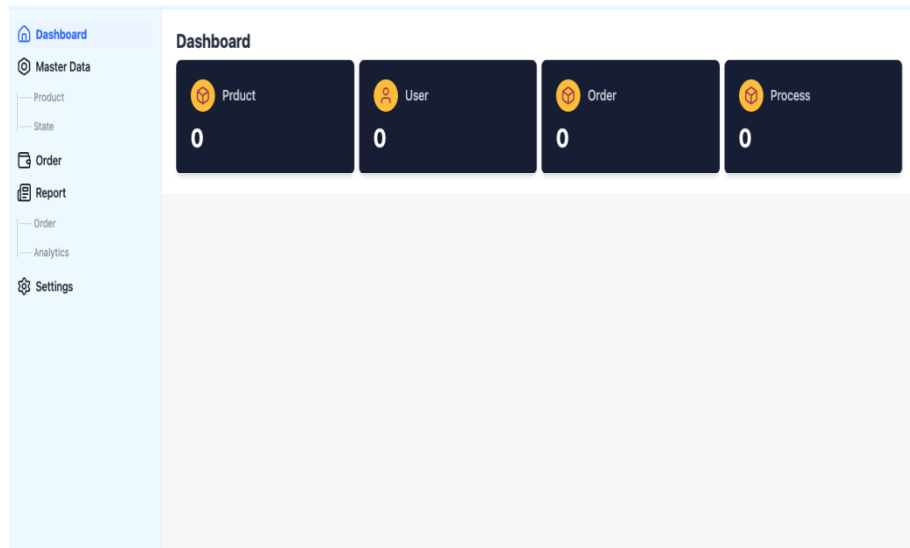
**Gambar 4. 11 Halaman Registrasi**



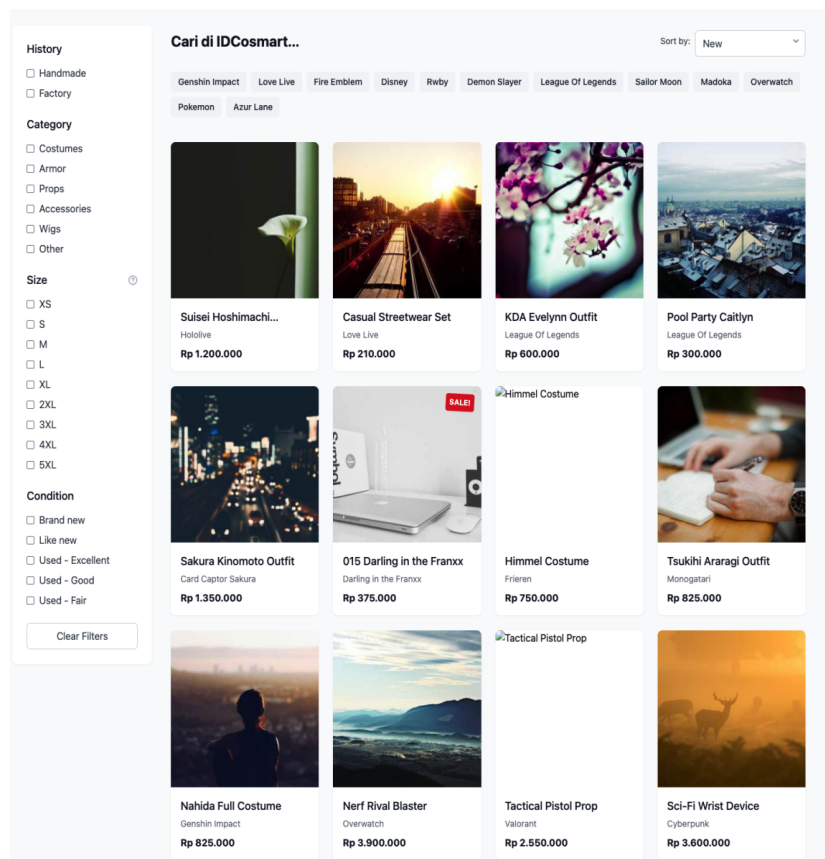
***Gambar 4. 12 Halaman Login***



***Gambar 4. 13 Halaman Beranda Pelanggan***

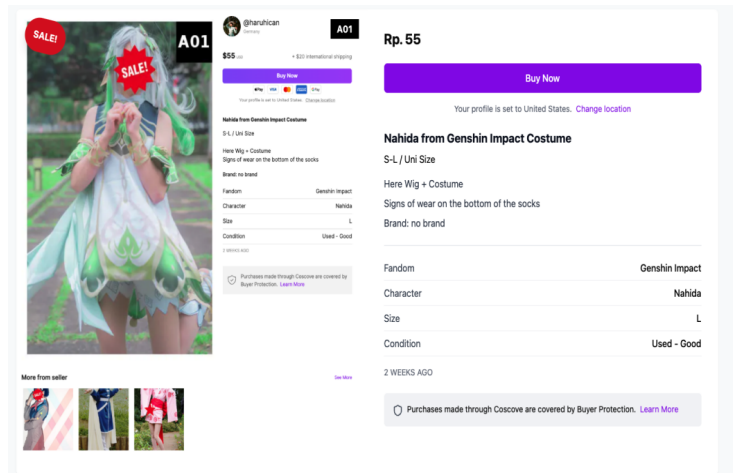


***Gambar 4. 14 Halaman Beranda Admin***

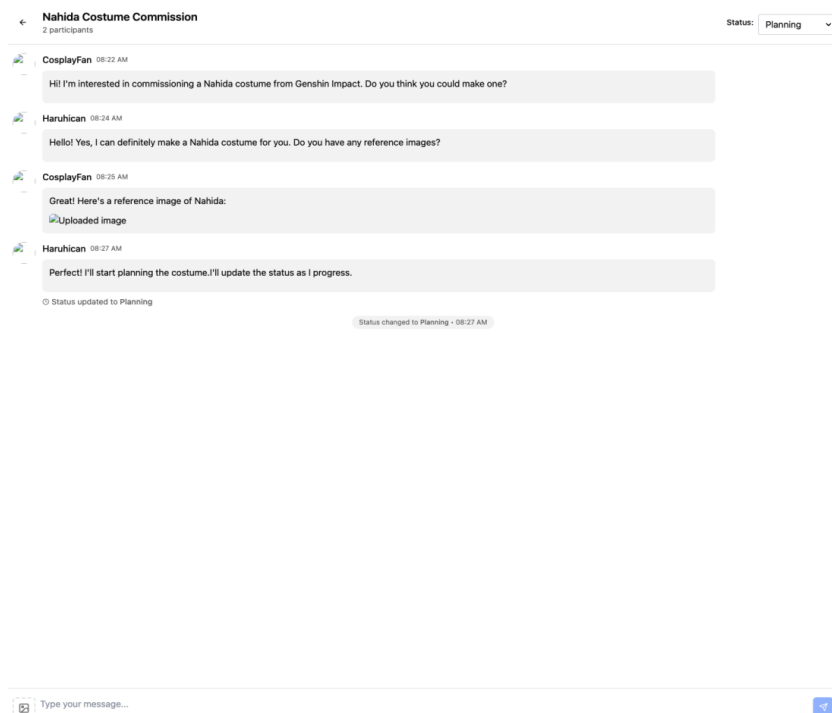


***Gambar 4. 15 Halaman Produk***

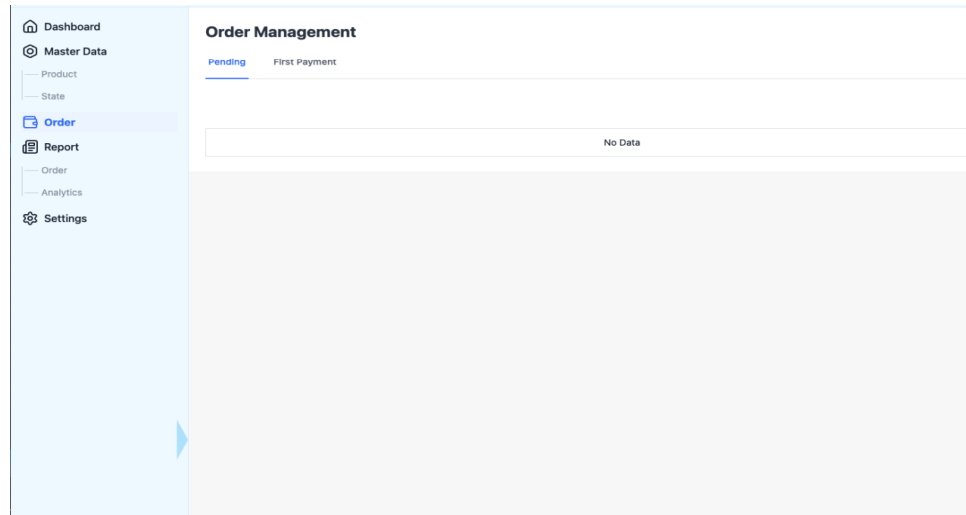




**Gambar 4. 16 Halaman Detail Produk**



**Gambar 4. 17 Halaman Pemesanan**



**Gambar 4. 18 Halaman Laporan Pemesanan Admin**

### 3. Desain Visual dan UX

Saya memberikan perhatian khusus pada aspek visual dan pengalaman pengguna:

- a. **Desain Bersih & Minimalis:** Saya memilih desain yang bersih dan minimalis untuk menonjolkan visual produk Mochi Studio yang artistik. Palet warna akan disesuaikan dengan branding Mochi Studio, kemungkinan kombinasi warna pastel atau cerah yang lembut, untuk menciptakan suasana yang *playful* namun tetap profesional dan elegan.
- b. **Navigasi Intuitif:** Navigasi dirancang agar sederhana dan mudah diakses. Saya akan menggunakan elemen seperti *sticky header* pada desktop dan menu *hamburger* pada perangkat *mobile* untuk mengarahkan pengguna dengan cepat ke halaman-halaman penting seperti Produk, Keranjang, Pesanan Saya, dan Akun.
- c. **Responsivitas:** Desain sistem ini akan sepenuhnya responsif, memastikan tampilan dan fungsionalitas optimal di berbagai jenis perangkat, mulai dari desktop, tablet, hingga smartphone. Hal ini menjamin pengalaman pengguna yang konsisten di mana pun mereka mengakses website.
- d. **Feedback Interaktif:** Saya mengimplementasikan berbagai bentuk *feedback* interaktif yang jelas kepada pengguna:

- e. Indikator *Loading*: Digunakan saat memuat data atau memproses transaksi, memberikan informasi visual bahwa sistem sedang bekerja.
- f. Validasi *Form*: Pesan kesalahan akan ditampilkan secara inline dan jelas saat pengguna mengisi form dengan data yang tidak valid, memandu mereka untuk koreksi.
- g. Visualisasi *Progress*: Untuk melacak orderProgress, saya akan menggunakan *progress bar* atau *timeline* visual pada halaman status pesanan. Ini memberikan representasi *real-time* tentang tahapan pengerjaan order, sesuai dengan state FSM yang sedang berjalan.

#### 4. Tools dan Implementasi

Dalam proses perancangan dan pengembangan UI:

- a. Desain UI/UX: Saya menggunakan *tools* seperti Figma untuk membuat *wireframe*, *mockup*, dan prototipe interaktif. Penggunaan *tool* ini memungkinkan saya untuk melakukan user *testing* awal dan iterasi desain secara efisien sebelum masuk ke tahap pengembangan kode.
- b. *Frontend Framework*: Implementasi *frontend* akan menggunakan *framework* JavaScript populer seperti React.js. *Framework* ini dipilih karena kemampuannya dalam menciptakan *declarative* UI dan pengelolaan *state* yang efisien, yang sangat cocok untuk berinteraksi dengan *backend* Node.js yang saya kembangkan.
- c. Komponen UI *Library*: Untuk mempercepat pengembangan dan menjaga konsistensi tampilan, saya akan memanfaatkan UI component library seperti Material-UI atau mengadopsi customizable CSS *framework* seperti Tailwind CSS. Ini menyediakan komponen siap pakai yang sudah teruji dan responsif.

## 5. Contoh Alur Pengguna (dengan FSM Implisit)

Berikut adalah alur pengguna dalam sistem pemesanan Mochi Studio, dengan implikasi FSM pada setiap tahapannya.

1. Pada sisi pelanggan :
  - a. Pengguna Membuka *Website*: Pengguna akan melihat *splash screen* Mochi Studio, kemudian secara otomatis diarahkan ke halaman beranda.
  - b. Login atau Registrasi: Pengguna akan masuk ke akun yang sudah ada atau mendaftar akun baru (User dibuat/diverifikasi).
  - c. Melihat Produk: Pengguna menjelajahi daftar product Mochi Studio, melihat detail produk, *productPrice*, dan opsi kustomisasi yang tersedia.
  - d. Memilih & Memesan Produk: Pengguna memilih produk yang diinginkan, menentukan jumlah (*qty*), dan menambahkannya ke *cart*.
  - e. Setelah selesai memilih, pengguna melanjutkan ke halaman *checkout*.
  - f. Proses *Checkout*: Pengguna mengkonfirmasi atau memilih alamat pengiriman (*userAddress*), memilih jasa pengiriman, dan memilih metode pembayaran.
  - g. Sistem kemudian membuat entitas order baru dengan *orderStatus* awal PENDING. Ini secara implisit memicu transition *CREATE\_ORDER* dalam FSM.
  - h. Pengguna melanjutkan proses pembayaran dan bisa melalui *send image* melalui *chat system*. Setelah pembayaran berhasil diverifikasi (*paymentStatus*: true dan entri payment baru tercatat), FSM akan memicu *event* *paymentDpReceived* atau *paymentFullReceived*, yang kemudian akan mengubah *orderStatus* menjadi *DP\_PAID* atau *PAID*.

- i. Ketika `orderStatus` telah mencapai PAID atau DP\_PAID, admin Mochi Studio akan memicu FSM `startProcessing`. Status order di sistem akan berubah menjadi PROCESSING.
- j. Pada halaman ini, pengguna akan melihat timeline `orderProgress` yang diperbarui secara berkala oleh admin (misalnya "Desain Dikerjakan", "Produksi Berlangsung", "Pengecekan Kualitas").
- k. Setelah semua tahapan pengerjaan diselesaikan, admin akan memicu FSM `markReadyForShipment`, mengubah `orderStatus` menjadi READY\_FOR\_SHIPMENT.
- l. Ketika admin menginput `awbShipment` dan memicu FSM `shipOrder`, `orderStatus` akan berubah menjadi SHIPPED. Pengguna kemudian dapat melacak status pengiriman melalui nomor resi yang diberikan.
- m. Setelah pesanan sampai di tangan pelanggan dan dikonfirmasi (baik oleh pelanggan atau melalui sistem pelacakan otomatis yang memicu FSM `confirmDelivery`), `orderStatus` akan berubah menjadi DELIVERED.
- n. Setelah periode tertentu (misalnya, 3 hari) atau setelah pengguna memberikan review produk (`isReview`), FSM `completeOrderProcess` akan dipicu, menandai `orderStatus` sebagai COMPLETED.

## 2. Manajemen Admin (Melalui Admin Dashboard/Panel):

Admin memiliki akses untuk melihat seluruh daftar order, mengubah `orderStatus` (secara langsung memicu transisi FSM), menginput `awbShipment`, dan memperbarui detail `orderProgress` untuk setiap pesanan. Selain itu, admin bertanggung jawab dalam pengelolaan product, `productPrice`, dan informasi *user* di dalam sistem.

Desain UI yang saya rancang, dengan integrasi erat bersama *Finite State Machine*, bertujuan tidak hanya untuk meningkatkan kenyamanan pelanggan dalam memesan produk Mochi Studio, tetapi juga untuk memberikan transparansi yang tinggi mengenai *progres* pesanan. Hal ini secara signifikan akan memudahkan manajemen operasional bagi pihak Mochi Studio, menciptakan sistem yang efisien dan *user-friendly*.

### 4.3. Code Generation

Dalam pengembangan aplikasi Mochi Studio yang menggunakan Node.js (Express.js) untuk backend dan Nuxt.js untuk *frontend*, *code generation* dapat secara signifikan mempercepat proses pengembangan, mengurangi kesalahan manual, dan memastikan konsistensi kode.

Berikut adalah gambaran dan rekomendasi terkait *code generation* untuk aplikasi Mochi Studio:

#### 4.3.1. Code Generation untuk Backend ( [Express.js](#) )

Untuk *backend* Mochi Studio, *code generation* dapat diterapkan pada berbagai lapisan:

##### a. Struktur Proyek Awal

Menggunakan *tools* seperti *express-generator* (atau *custom script* yang dibuat sendiri) untuk membuat struktur folder dasar proyek Express.js, termasuk direktori untuk *routes*, *controllers*, *models*, *middlewares*, dan konfigurasi.

##### b. API Endpoints

Membuat *script* atau menggunakan *library* yang dapat meregenerasi *boilerplate* untuk *routes* (misalnya, GET */api/items*, POST */api/items*, PUT */api/items/:id*, DELETE */api/items/:id*). Secara otomatis membuat *controller* yang sesuai dengan fungsi CRUD (Create, Read, Update, Delete) untuk setiap entitas (misalnya, *itemController.js*, *orderController.js*, *userController.js*). Ini dapat mencakup template dasar untuk validasi *input* dan penanganan respons.

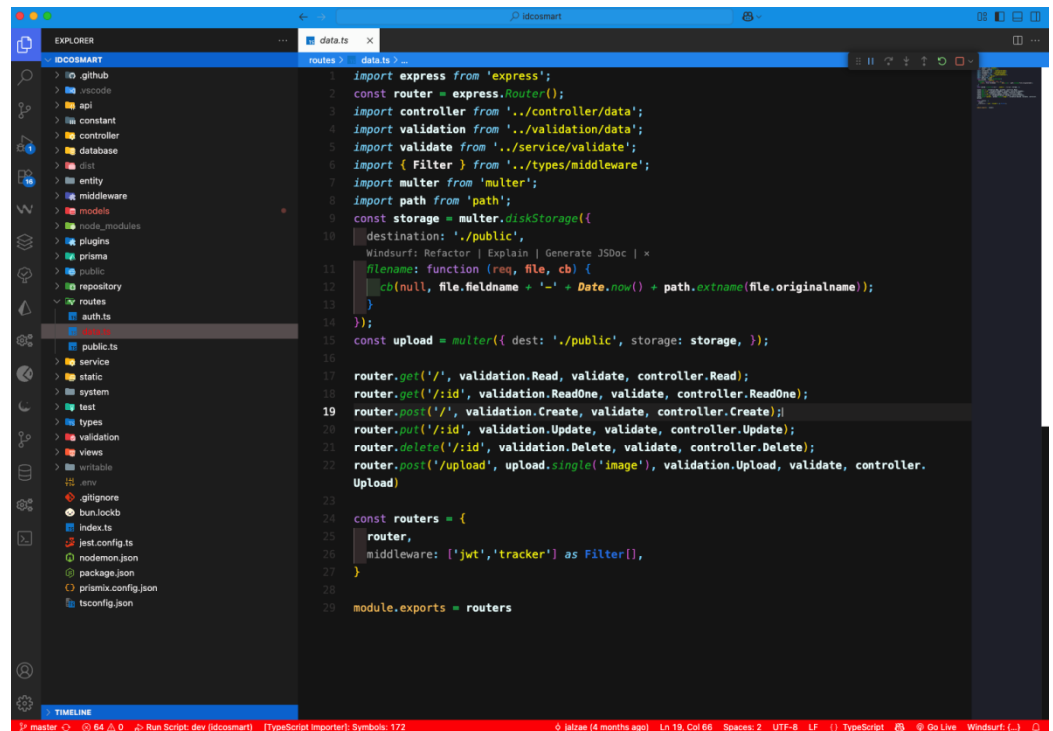
##### c. Model Database

Jika menggunakan ORM (*Object-Relational Mapping*) seperti Sequelize (untuk MySQL/PostgreSQL) atau Mongoose (untuk

MongoDB), *code generation* dapat digunakan untuk membuat definisi model berdasarkan skema *database*.

#### d. Autentikasi dan Otorisasi

Menggenerasi *boilerplate* untuk *middleware* autentikasi (misalnya, verifikasi JWT) dan *middleware* otorisasi (misalnya, pengecekan peran pengguna).



Gambar 4. 19 Struktur Express Js

#### 4.3.2. Code Generation untuk Frontend ([Nuxt.js](#))

Nuxt.js sendiri sudah memiliki banyak fitur *code generation* bawaan yang sangat membantu:

##### a. Scaffolding Proyek [Nuxt.js](#)

Perintah `npx nuxi init <project-name>` akan menginisialisasi proyek Nuxt.js dengan struktur folder standar (pages, components, layouts, assets, public, server, dll.).

##### b. Generasi Halaman (*Pages*)

Nuxt.js secara otomatis membuat routing berdasarkan struktur *folder pages*. Namun, *custom script* dapat dibuat untuk menggenerasi boilerplate halaman dengan template dasar (misalnya, halaman daftar produk, halaman detail produk, halaman keranjang).

### c. Generasi Komponen (*Components*)

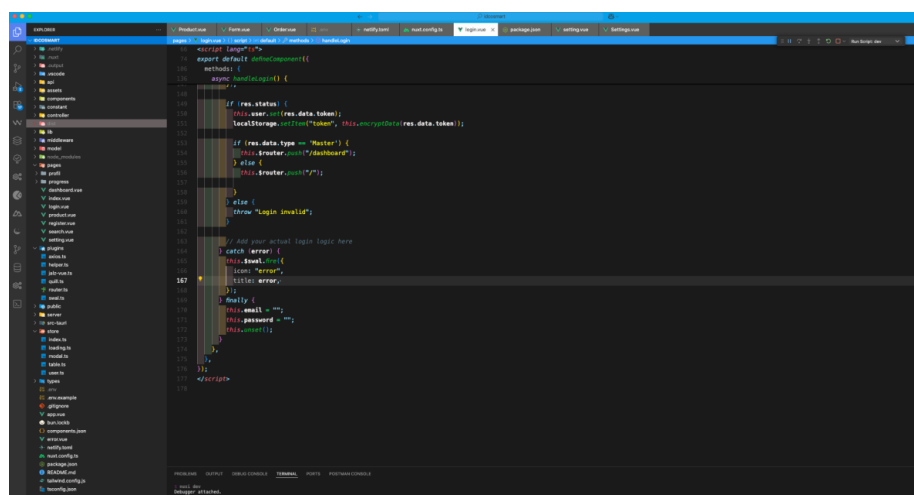
Membuat *boilerplate* untuk komponen Vue.js yang sering digunakan (misalnya, `ProductCard.vue`, `OrderItem.vue`, `NotificationToast.vue`). Ini dapat mencakup template dasar dengan props dan emits yang umum.

### d. Integrasi API Services

Menggenerasi template untuk file layanan API yang akan berinteraksi dengan *backend* Express.js. Misalnya, `api/products.js` yang berisi fungsi untuk `getProducts()`, `createProduct()`, dll.

### e. State Management (Pinia)

Menggenerasi *boilerplate* untuk store Pinia, termasuk definisi state, getters, actions, dan mutations (jika menggunakan Vuex Options API).



Gambar 4. 20 Struktur NuxtJs



#### 4.3.3. Integrasi dengan *Tools* Otomasi (VS Code & Windsurf)

Untuk lebih mengoptimalkan proses *code generation* dan otomasi, integrasi dengan *Integrated Development Environment* (IDE) dan *tools* pihak ketiga sangat direkomendasikan:

##### 1. Visual Studio Code (VS Code)

- a. *Snippet Extensions*: Membuat *custom snippets* di VS Code untuk blok kode yang sering digunakan. Misalnya, *snippet* untuk membuat controller Express.js baru atau komponen Nuxt.js dengan struktur dasar.
- b. *Task Automation*: Mengkonfigurasi *tasks.json* di VS Code untuk menjalankan *script code generation* secara langsung dari IDE. Contohnya, menjalankan perintah `npm run generate:model <modelName>` yang akan memicu *script* untuk membuat model baru.
- c. *Extensions*: Memanfaatkan ekstensi VS Code yang mendukung *code generation* atau *scaffolding* untuk Node.js/Express.js dan Vue/Nuxt.js.
- d. Windsurf (atau *Tools* Serupa untuk Otomasi Alur Kerja):

Meskipun "Windsurf" mungkin bukan *tool code generation* yang umum, konsep *tool* otomasi alur kerja (seperti task runners atau build automation tools) sangat relevan.

- e. *Custom Scripting*: Mengembangkan *custom script* (misalnya dengan Node.js sendiri, Python, atau shell scripts) yang dapat diintegrasikan ke dalam alur kerja CI/CD (*Continuous Integration/Continuous Deployment*) atau dijalankan secara lokal.

Manfaat *Code Generation* untuk Mochi Studio yang didapatkan sebagai berikut :

- a. Percepatan Pengembangan: Mengurangi waktu yang dihabiskan untuk menulis kode berulang.
- b. Konsistensi Kode: Memastikan semua bagian aplikasi mengikuti standar dan pola yang sama.
- c. Mengurangi Kesalahan: Mengeliminasi kesalahan ketik dan kesalahan manual yang sering terjadi saat menulis kode berulang.
- d. Fokus pada Logika Bisnis: Pengembang dapat lebih fokus pada implementasi logika bisnis inti daripada *boilerplate*.
- e. Memudahkan *Onboarding*: Pengembang baru dapat lebih cepat memahami struktur proyek dan mulai berkontribusi.

Dengan memanfaatkan *code generation* secara efektif, Mochi Studio dapat membangun sistem pemesanan barang yang lebih cepat, lebih andal, dan lebih mudah dikelola.

#### 4.4. *Testing*

Proses pengujian sistem akan dilakukan secara menyeluruh untuk memastikan kualitas dan keandalan. Pendekatan *Gray Box Testing* akan digunakan, menggabungkan pemahaman tentang struktur internal kode (seperti alur *Finite State Machine*) dengan pengujian fungsional dari sudut pandang pengguna. Pengujian akan mencakup beberapa aspek utama:

1. ***Unit Testing***: Menguji komponen atau modul kode secara individual untuk memastikan setiap bagian kecil berfungsi dengan benar sebelum diintegrasikan.
2. ***Automated Testing***:
  - a. ***API Testing***: Menguji *endpoint API backend* yang dibangun dengan Express.js untuk memastikan data dikirim dan diterima dengan benar, serta logika bisnis berjalan sesuai harapan.
  - b. ***Frontend Testing***: Menguji antarmuka pengguna yang dibangun dengan Nuxt.js untuk memastikan interaksi elemen, tampilan, dan alur pengguna berjalan mulus tanpa bug.

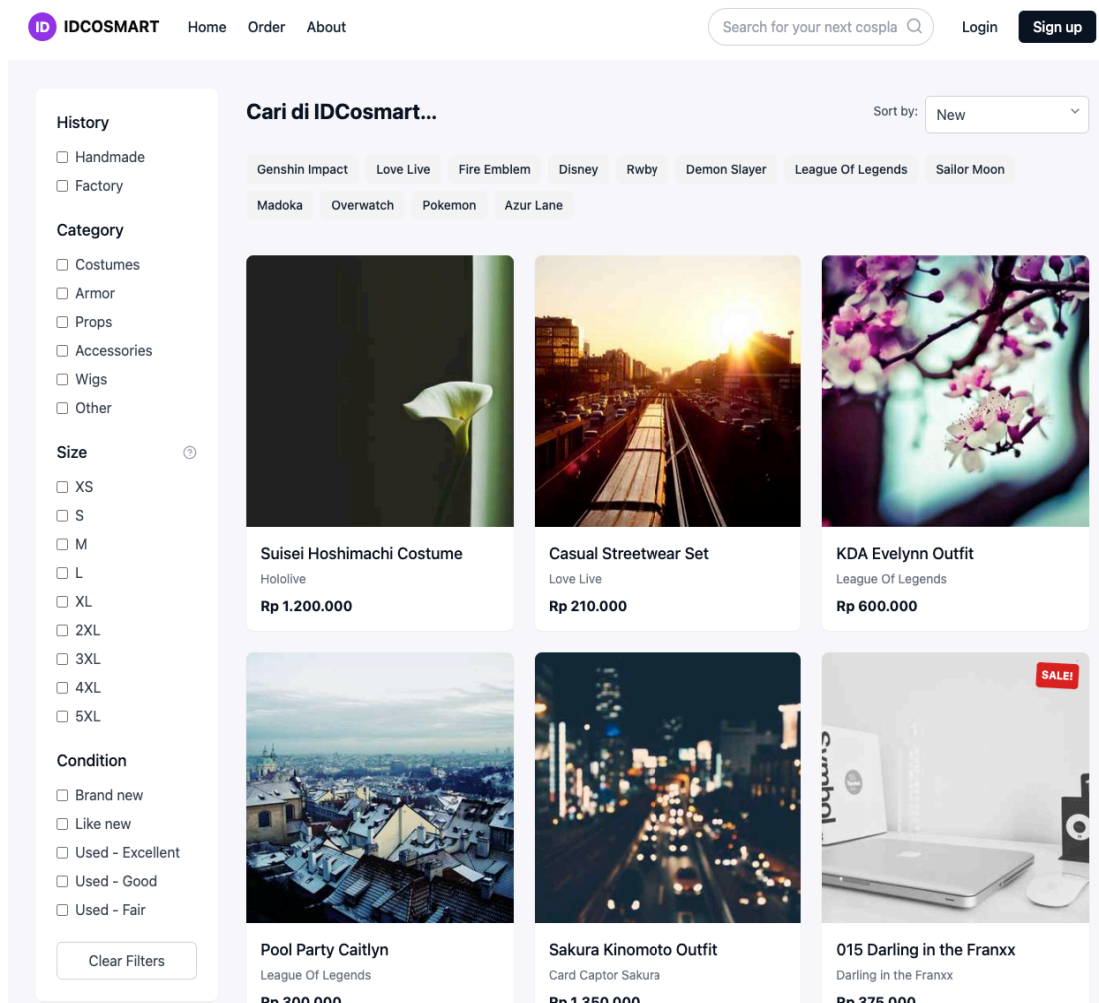
3. **Functional Testing:** Memastikan semua fitur utama berjalan sesuai spesifikasi, mulai dari proses pendaftaran dan *login* pelanggan, pengunggahan desain, pemilihan spesifikasi produk, hingga perubahan status pesanan oleh admin dan notifikasi kepada pelanggan.
4. **Usability Testing:** Menguji kemudahan penggunaan *website* oleh pelanggan dan admin Mochi Studio, memastikan antarmuka yang intuitif dan navigasi yang lancar agar pengguna dapat menyelesaikan tugas dengan efisien.
5. **Performance Testing:** Menguji kemampuan sistem untuk menangani volume pesanan yang tinggi atau banyak pengguna yang mengakses secara bersamaan, memastikan Node.js memberikan respons yang cepat tanpa lag.
6. **Security Testing:** Memastikan data pelanggan dan transaksi pesanan aman dari potensi ancaman siber seperti SQL Injection, Cross-Site Scripting (XSS), atau serangan lainnya, melindungi privasi dan integritas data.
7. **Compatibility Testing:** Menguji sistem di berbagai browser web dan perangkat (desktop, tablet, smartphone) untuk menjamin pengalaman pengguna yang optimal di mana pun sistem diakses.

Pengujian yang komprehensif ini bertujuan untuk memastikan bahwa Sistem Pemesanan Cetak Tekstil Mochi Studio dapat beroperasi secara stabil dan andal. Dengan demikian, sistem akan mendukung kegiatan operasional Mochi Studio secara lebih efisien, mengurangi kesalahan manual, dan meningkatkan kepuasan pelanggan melalui layanan pemesanan yang mudah, cepat, dan transparan.

## 4.5. Support

### 4.5.1 Publikasi Web

Sistem Pemesanan Barang yang dikembangkan untuk Mochi Studio merupakan sebuah aplikasi berbasis *website* yang dirancang menggunakan teknologi Node.js untuk mendukung operasional studio secara efisien dan terintegrasi. Nantinya website ini akan di *deploy* pada alamat *idcosmart.netlify.app*.



Gambar 4. 21 Publikasi Web di Netlify

### 1. Deskripsi Sistem

- a. Sistem ini memfasilitasi proses pengelolaan pesanan barang, pemantauan status order secara *real-time*, dan pemesanan barang oleh pelanggan maupun admin studio. Dengan fitur yang mudah digunakan dan berbasis *web*, sistem ini memungkinkan

akses kapan saja dan dari mana saja tanpa perlu instalasi aplikasi tambahan, mendukung fleksibilitas operasional Mochi Studio.

## 2. Teknologi yang Digunakan

- a. **Backend:** Node.js dengan *framework* Express.js sebagai *server-side runtime environment* yang handal dan scalable.
- b. **Frontend:** Nuxt.js (berbasis Vue.js) untuk antarmuka pengguna yang responsif, dinamis, dan memberikan pengalaman terbaik di berbagai perangkat.
- c. **Database:** MongoDB/MySQL (tergantung arsitektur yang dipilih) untuk penyimpanan data barang, transaksi pesanan, dan pengguna.
- d. **API:** RESTful API untuk komunikasi yang efisien dan terstruktur antara *frontend* dan *backend*.
- e. **Deployment:** *Backend* akan di-*deploy* pada Ubuntu Server untuk stabilitas dan kontrol penuh, sementara *frontend* (Nuxt.js) akan di-*deploy* menggunakan Netlify untuk *hosting* yang cepat, aman, dan mudah dikelola, memastikan aksesibilitas luas oleh pelanggan dan tim Mochi Studio.

## 3. Manfaat Publikasi Sistem Web

Pengelolaan Data Lebih Efisien: Mengotomatiskan proses pemantauan pesanan dan pengelolaan barang yang sebelumnya manual, mengurangi beban kerja administratif.

- a. Kemudahan Akses: Pelanggan dan admin dapat dengan mudah melakukan pemesanan dan memantau status order dari perangkat apapun yang terhubung internet.
- b. Transparansi dan Akurasi: Meminimalisir kesalahan data pesanan dan status transaksi yang dapat terjadi pada proses manual, meningkatkan kepercayaan pelanggan.

- c. Peningkatan Layanan: Memberikan respon cepat terhadap permintaan dan pemesanan pelanggan melalui fitur pemesanan *online* dan pelacakan status *real-time*.
- d. Skalabilitas: Sistem berbasis Node.js memungkinkan penambahan fitur dan pengguna seiring perkembangan bisnis Mochi Studio di masa mendatang.

#### 4. Proses Publikasi

Proses publikasi website sistem ini terdiri dari 2 aplikasi terpisah antara *frontend* dan *backend*. Pada publikasi *backend* akan di tempatkan pada repository kemudian *dcompile* pada server serta *dirunning* menggunakan *Process Manager*. Pada bagian *frontend* akan langsung di *deploy* pada hosting jamstick seperti keterangan pada *deploy* diagram.

##### 4.5.2 Spesifikasi Hardware dan Software

Spesifikasi hardware dan software untuk sistem pemesanan print textile dengan model *Finite State Machine* untuk Mochi Studio diperkirakan sebagai berikut :

Tabel IV. 2 Tabel Spesifikasi

Aspek	Spesifikasi Hardware	Spesifikasi Software
<b>Server</b>	<ul style="list-style-type: none"> <li>- CPU minimal quad-core (contoh: AMD Ryzen / Intel Core i5 ke atas) untuk menangani banyak request secara simultan</li> <li>- RAM minimal 8 GB agar mendukung proses Node.js yang optimal</li> <li>- Media penyimpanan SSD untuk akses data yang cepat</li> </ul>	<ul style="list-style-type: none"> <li>- Node.js (<i>runtime environment</i> berbasis JavaScript menggunakan V8 engine)</li> <li>- Modul <i>clustering</i> Node.js untuk meningkatkan performa dan distribusi beban CPU</li> <li>- Database seperti MongoDB atau MySQL untuk penyimpanan data stok, pengguna, dan transaksi</li> <li>- Web server seperti Nginx atau</li> </ul>

		Apache sebagai reverse proxy dan load balancer pada deployment produksi
<b>Client</b>	- Perangkat desktop/laptop atau smartphone dengan kemampuan akses browser internet	- Browser modern (Google Chrome, Mozilla Firefox, Microsoft Edge, dll.) yang mendukung HTML5, CSS3, dan JavaScript
<b>Jaringan</b>	- Koneksi internet stabil, minimal broadband untuk akses web secara real-time	- Protokol HTTP/HTTPS untuk akses website, serta implementasi keamanan SSL/TLS agar data transaksi terproteksi

#### 4.6. Spesifikasi Dokumen Sistem Usulan

Dokumen spesifikasi sistem merupakan dokumen yang berisi nantinya bagaimana sistem dibuat dan berjalan. Ini akan menjadi penting untuk pengguna dalam menjalankan dan mengembangkan sistem. Dengan adanya sistem ini maka berikut urutan dokumen sistem yang ada pada sistem sebagai berikut :

##### 1. Pendahuluan

- a. **Latar Belakang:** Dokumen ini menjelaskan kebutuhan Mochi Studio untuk beralih dari proses pengelolaan pesanan dan barang yang masih konvensional (menggunakan WhatsApp dan Facebook sebagai catatan order manual) menuju sistem yang terdigitalisasi. Proses manual ini

seringkali menimbulkan tantangan dalam efisiensi, akurasi data, dan pelacakan status pesanan secara real-time.

- b. **Tujuan Sistem:** Tujuan utama pengembangan sistem ini adalah untuk meningkatkan efisiensi pengelolaan pesanan dan barang, mempercepat proses transaksi, menyediakan pelacakan status pesanan yang transparan menggunakan model *Finite State Machine* (FSM), serta mendukung pengambilan keputusan yang lebih baik bagi Mochi Studio.
- c. **Ruang Lingkup Sistem:** Sistem ini akan mencakup fungsi-fungsi utama seperti pendaftaran dan manajemen profil pelanggan, katalog barang, proses pemesanan online, pelacakan status pesanan, manajemen transaksi oleh admin, serta pembuatan laporan. Aktor utama yang akan berinteraksi dengan sistem adalah Admin Mochi Studio dan Pelanggan.

## 2. Analisis Sistem

- a. **Analisis Sistem Eksisting:** Saat ini, Mochi Studio mengandalkan komunikasi langsung melalui WhatsApp dan Facebook untuk menerima pesanan. Pencatatan pesanan dan detail barang dilakukan secara manual, yang rentan terhadap kesalahan, duplikasi data, dan kesulitan dalam memantau kemajuan setiap order. Proses pengerjaan order juga masih manual dan belum terintegrasi dengan sistem pencatatan.
- b. **Elisitasi Kebutuhan:** Pengumpulan kebutuhan fungsional dan non-fungsional akan dilakukan melalui wawancara dengan staf Mochi Studio dan observasi alur kerja yang ada. Kebutuhan ini kemudian akan dikelompokkan dalam tahap-tahap elisitasi untuk memastikan semua aspek tercakup hingga draf final.
- c. **Diagram UML:** Diagram UML (*Unified Modeling Language*) seperti *Use Case Diagram* dan *Activity Diagram* akan digunakan untuk merepresentasikan secara visual proses bisnis yang diusulkan dan interaksi antara pengguna dengan sistem.



### 3. Perancangan Sistem

- a. **Spesifikasi Fungsional:** Sistem akan memiliki fungsi utama meliputi: pendaftaran dan manajemen data pelanggan, pengelolaan katalog barang (penambahan, perubahan, penghapusan item), fitur pemesanan online oleh pelanggan, pelacakan status pesanan menggunakan model Finite State Machine (misalnya, dari "Menunggu Pembayaran" hingga "Selesai"), manajemen transaksi oleh admin (konfirmasi, pembatalan, pembaruan status), serta pembuatan laporan penjualan dan produksi.
- b. **Spesifikasi Non-Fungsional:** Sistem akan dirancang untuk memiliki performa yang optimal, keamanan data dan transaksi yang kuat, antarmuka yang user-friendly dan intuitif, serta platform berbasis website yang responsif.
- c. **Desain Basis Data:** Definisi tabel dan relasi antar tabel akan dirancang untuk menyimpan data pengguna, data barang, data pesanan, status transaksi, dan informasi terkait lainnya. Desain ini akan mengacu pada model data fisik yang valid untuk memastikan integritas dan efisiensi query.
- d. **Arsitektur Sistem:** Sistem akan mengadopsi arsitektur *client-server*. Bagian *backend* akan dibangun menggunakan Node.js dengan framework Express.js untuk menangani logika bisnis dan API. Bagian frontend akan dikembangkan menggunakan Nuxt.js (berbasis Vue.js) untuk antarmuka pengguna yang dinamis dan responsif. MySQL atau PostgreSQL akan digunakan sebagai sistem manajemen database relasional.
- e. **Desain Antarmuka:** *Mockup* antarmuka pengguna akan dibuat untuk memvisualisasikan tata letak, elemen interaktif, dan alur navigasi, memastikan kemudahan akses dan pengalaman pengguna yang optimal.

### 4. Implementasi dan Pengujian

- a. **Metode Implementasi:** Tahapan pengembangan perangkat lunak akan melibatkan *coding* menggunakan Node.js (Express.js) dan Nuxt.js,

diikuti dengan deployment pada server hosting dengan protokol keamanan (HTTPS) untuk akses yang aman.

- b. **Pengujian Sistem:** Pengujian akan dilakukan secara komprehensif menggunakan metode *Gray Box Testing*, yang mencakup pemahaman internal kode dan pengujian dari sudut pandang pengguna. Detail rencana pengujian meliputi:
- c. **Unit Testing:** Menguji komponen kode secara individual.
- d. **Automated Testing:** Baik untuk API backend maupun tampilan *frontend*.
- e. **Functional Testing:** Memastikan semua fitur berjalan sesuai spesifikasi.
- f. **Performance Testing:** Menguji ketahanan sistem di bawah beban tinggi.
- g. **Security Testing:** Mengidentifikasi dan mengatasi kerentanan keamanan.
- h. **Compatibility Testing:** Memastikan sistem berfungsi di berbagai *browser* dan perangkat.

## 5. Jadwal dan Estimasi Biaya

- a. **Jadwal Implementasi:** Rencana *timeline* pengerjaan akan disusun, mencakup fase analisis, desain, *coding*, pengujian, hingga *deployment* dan implementasi.
- b. **Estimasi Biaya:** Perkiraan biaya yang diperlukan akan dihitung, meliputi perangkat keras (jika ada), perangkat lunak (lisensi jika diperlukan), biaya pengembangan, dan pemeliharaan.

## 6. Dokumentasi Pendukung

- a. **Manual Pengguna:** Panduan lengkap akan disediakan untuk membantu admin Mochi Studio dan pelanggan dalam menggunakan sistem.

- b. **Dokumentasi Teknis:** Dokumen ini akan merinci struktur kode, skema database, dan konfigurasi server untuk memudahkan pemeliharaan dan pengembangan di masa depan.

Dokumen sistem usulan ini berfungsi sebagai acuan pengembangan sistem pemesanan barang berbasis *website* dengan Node.js, Express.js, dan Nuxt.js untuk Mochi Studio. Diharapkan sistem ini dapat menyederhanakan proses pengelolaan pesanan dan barang secara efektif, akurat, dan terintegrasi, serta meningkatkan kepuasan pelanggan melalui layanan yang lebih modern dan efisien.

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Berdasarkan hasil penelitian dan pengembangan sistem *e-commerce* pemesanan berbasis *Finite State Machine* (FSM), dapat disimpulkan bahwa penerapan FSM secara efektif berhasil memberikan alur kontrol yang sistematis, terstruktur, dan mudah dipelihara. FSM mampu mengelola status pengguna dan pesanan dengan jelas, mulai dari proses *login*, pemilihan produk, *checkout*, pembayaran, hingga konfirmasi dan pengiriman pesanan. Sistem ini menunjukkan peningkatan efisiensi, keandalan, dan keteraturan dalam proses pemesanan dibandingkan metode manual sebelumnya.

Pencapaian tujuan sistem dapat dirangkum sebagai berikut:

1. Sistem pemesanan digital berhasil dibangun dan menggantikan metode manual via WhatsApp, sehingga seluruh proses kini lebih terorganisir, terdokumentasi, dan profesional.
2. Penerapan model *Finite State Machine* terbukti berhasil memastikan setiap pesanan mengikuti *workflow* yang sistematis, meminimalkan kesalahan alur seperti kehilangan pesanan atau duplikasi status.
3. Sistem pencatatan status pemesanan telah dikembangkan, sehingga pengguna dan admin dapat memantau setiap tahapan produksi secara *real-time*, meningkatkan transparansi dan kontrol proses.
4. Fitur pelacakan pesanan dapat lebih transparan, memungkinkan pelanggan memantau status pengiriman dengan mudah tanpa perlu konfirmasi manual.
5. Manajemen data pelanggan kini lebih terstruktur, membuka peluang untuk penerapan strategi promosi berbasis data pemesanan yang lebih akurat dan tertarget.

Meskipun demikian, sistem masih memiliki beberapa keterbatasan, seperti belum tersedianya fitur pelacakan riwayat transisi status secara menyeluruh, absennya notifikasi otomatis pada perubahan status pesanan, dan keterbatasan integrasi sistem

pembayaran. Hal ini dapat menjadi fokus untuk pengembangan lebih lanjut pada tahap berikutnya.

## 5.2. Saran

Agar sistem pemesanan berbasis FSM ini dapat dikembangkan lebih baik di masa depan, beberapa saran yang dapat diberikan adalah:

1. Menambahkan sistem log dan riwayat transisi status, agar pengembang atau admin dapat melakukan audit dan pelacakan proses transaksi dengan lebih akurat.
2. Mengintegrasikan sistem notifikasi otomatis, baik melalui email atau WhatsApp API, agar pengguna mendapatkan informasi *real-time* mengenai status pesanan mereka.
3. Meningkatkan fleksibilitas FSM dengan mendukung kondisi dinamis, seperti transaksi dengan banyak skenario (diskon, retur, pembatalan sebagian).
4. Melakukan pengujian sistem dengan skenario *edge-case*, misalnya saat server sibuk, terjadi error saat pembayaran, atau ada pesanan berulang dari pengguna yang sama.

Dengan pengembangan lebih lanjut berdasarkan saran tersebut, diharapkan sistem pemesanan yang dibangun tidak hanya stabil dan efisien, tetapi juga adaptif terhadap kebutuhan pengguna dan skenario transaksi yang semakin kompleks di masa mendatang.

## DAFTAR PUSTAKA

- [1] R. A. Pratama, A. Nugroho, dan M. R. Lestari, “Perancangan Sistem Informasi Pemesanan Berbasis Web pada Layanan Digital Printing Medina Printing Menggunakan Metode Rapid Application Development (RAD),” *Jurnal Sistem Informasi dan Teknologi*, vol. 8, no. 2, pp. 101–110, 2025.
- [2] S. Agustin, D. H. Purnomo, dan R. Yuliana, “Penerapan Metode Finite State Machine (FSM) dalam Permainan Tradisional Setatak Berbasis Android,” *Jurnal Teknologi Informasi dan Komputer*, vol. 7, no. 1, pp. 45–52, 2021.
- [3] Rohi Abdulloh, *7 IN 1 PEMROGRAMAN WEB UNTUK PEMULA*. PT Elex Media Komputindo, Jakarta, 2018.
- [4] Didik Setiawan, *BUKU SAKTI PEMROGRAMAN WEB: HTML, CSS, PHP, MySQL & JAVASCRIPT*. START UP, 2018.
- [5] M. Marimin, H. Tanjung, and H. Prabowo, *Sistem Informasi Manajemen Sumber Daya Manusia*. Jakarta: Grasindo, 2009.
- [6] G. Patria, “Aplikasi Booking Fitting Pakaian pada Toko Busana Nabeel Textile Berbasis Web,” *Jurnal Ilmu Komputer*, Volume 7 Nomor 2 Oktober 2021
- [7] F. Wagner, R. Schmuki, T. Wagner, dan P. Wolstenholme, *Modeling Software with Finite State Machines: A Practical Approach*, 2nd ed. Boca Raton, FL: Auerbach Publishers, 2019.
- [8] Ahmad, N., Krisnanik, E., Rupiele, F. G. J., Muliawati, A., Syamsiyah, N., Kraugusteeliana, K., Cahyono, B. D., Sriyeni, Y., Kristanto, T., Irwanto, I., & Guntoro, G. (2022). *Analisa & perancangan sistem informasi berorientasi objek*. Indonesia: Penerbit Widina.

## DAFTAR RIWAYAT HIDUP

### I. Biodata Mahasiswa

NIM : 12240135  
Nama Lengkap : Jaisy Al Huda Shibghotulloh  
Tempat/Tanggal Lahir : Tegal, 10 November 1995  
Alamat Lengkap : Jl. Arjuna No. 113 , Kota Tegal. Provinsi Jawa Tengah

### II. Pendidikan

1. SMKN 2 Kota Tegal lulus tahun 2012
2. D3 Teknologi Komputer AMIK BSI Purwokerto, lulus tahun 2015

### III. Riwayat pengalaman berorganisasi / pekerjaan

1. Technical Support di PT. Roda Chakra Kencana tahun 2021 s.d 2022
2. Backend Developer di PT. Mahkota Giri Suprana tahun 2021 s.d 2023
3. Instruktur Software Developer di LPK Arum tahun 2018 s.d sekarang
4. Software Developer di Yayasan Ribathul Ukhuwah tahun 2019 s.d sekarang
5. Software Developer di PT Ebas Solusi Teknologi tahun 2023 s.d sekarang
6. Instruktur Kelas Komputer di Universitas Tujuh Belas Agustus tahun 2025 s.d sekarang
7. Front End Developer di Kemang Internet (SG)



Jakarta, 28 Juli 2025









**Jaisy Al Huda Shibghotulloh**



## LEMBAR BIMBINGAN TUGAS AKHIR

### UNIVERSITAS NUSA MANDIRI

NIM : 12240135  
 Nama Lengkap : Jaisy Al Huda Shibghotulloh  
 Dosen Pembimbing : Astriana Mulyani, S.Si, M.Kom  
 Judul Tugas Akhir : Sistem Pemesanan Print Tekstil Dengan Model Finite State Machine

No	Tanggal Bimbingan	Pokok Bahasan	Paraf Dosen Pembimbing
1	5 Juni 2025	Pengajuan Judul	
2	9 Juni 2025	ACC Judul dan pengajuan Bab I	
3	18 Juni 2025	Revisi Bab I dan pengajuan Bab II	
4	27 Juni 2025	ACC Bab I dan revisi Bab II	
5	7 Juli 2025	ACC Bab II dan pengajuan Bab III	
6	16 Juli 2025	Revisi Bab III dan pengajuan Bab IV	
7	22 Juli 2025	ACC Bab III, revisi Bab IV, dan pengajuan Bab V	
8	28 Juli 2025	Demonstrasi program dan ACC keseluruhan	

Catatan untuk Dosen Pembimbing  
Bimbingan Tugas Akhir

- Dimulai pada tanggal : 5 Juni 2025
- Diakhiri pada tanggal : 28 Juli 2025
- Jumlah pertemuan bimbingan : 8 Pertemuan

Disetujui oleh,  
Dosen Pembimbing



(Astriana Mulyani, S.Si, M.Kom)



# Mochi Empire

+62 896-6610-0887

empiremochi@gmail.com

s.id/mochiempire



## SURAT KETERANGAN RISET

Yang bertanda tangan di bawah ini:

Nama : Jaisy Al Huda Shibghotulloh  
NIM : 12240135  
Program Studi : Informatika  
Fakultas : Teknologi Informasi  
Perguruan Tinggi : Universitas Nusa Mandiri

Dengan ini menerangkan bahwa mahasiswa tersebut di atas benar sedang melakukan kegiatan riset sebagai bagian dari penyusunan Tugas Akhir dengan judul:

### **“Sistem Pemesanan Print Tekstil Dengan Model Finite State Machine”**

Penelitian ini dilaksanakan secara mandiri oleh mahasiswa yang bersangkutan dan menjadi bagian dari pemenuhan syarat akademik untuk memperoleh gelar Sarjana Komputer (S.Kom) di Universitas Nusa Mandiri.

Surat keterangan ini dibuat untuk digunakan sebagaimana mestinya dan dapat dipergunakan sebagai bukti kegiatan riset mahasiswa.

Demikian surat keterangan ini dibuat dengan sebenar-benarnya.

Tegal, 28 Juli 2025

**Mochi Studio**

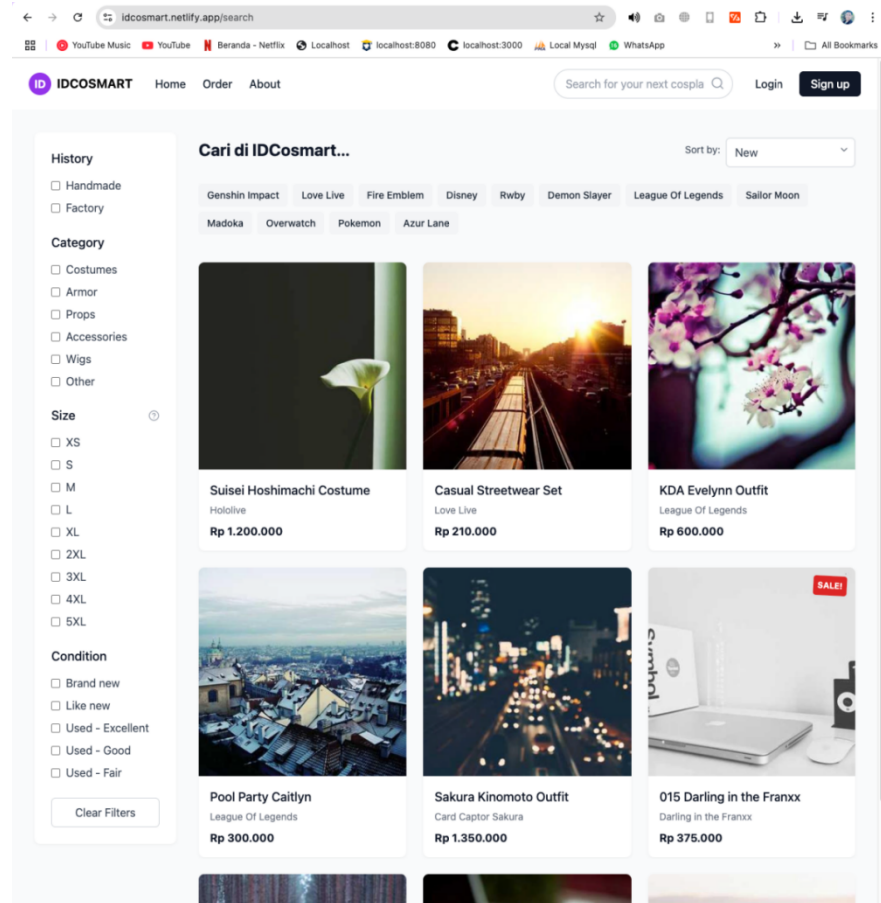
Anggun Intan N. A.

## LAMPIRAN

### A. Dokumen Sistem Berjalan



## B. Dokumen Sistem Usulan



## C. Bukti Hasil Pengecekan Plagiarisme

### 1. Bab I



## Plagiarism Checker X - Report

Originality Assessment

**0%**

Overall Similarity

**Date:** Aug 3, 2025  
**Matches:** 0 / 154 words  
**Sources:** 0

**Remarks:** No similarity found,  
your document looks healthy.

**Verify Report:**  
Scan this QR Code



### 2. Bab II



## Plagiarism Checker X - Report

Originality Assessment

**0%**

Overall Similarity

**Date:** Aug 3, 2025  
**Matches:** 0 / 168 words  
**Sources:** 0

**Remarks:** No similarity found,  
your document looks healthy.

**Verify Report:**  
Scan this QR Code



### 3. Bab III



## Plagiarism Checker X - Report

Originality Assessment

**0%**

**Overall Similarity**

**Date:** Aug 3, 2025  
**Matches:** 0 / 71 words  
**Sources:** 0

**Remarks:** No similarity found,  
your document looks healthy.

**Verify Report:**  
Scan this QR Code



### 4. Bab IV



## Plagiarism Checker X - Report

Originality Assessment

**0%**

**Overall Similarity**

**Date:** Aug 3, 2025  
**Matches:** 0 / 1053 words  
**Sources:** 0

**Remarks:** No similarity found,  
your document looks healthy.

**Verify Report:**  
Scan this QR Code



### 5. Bab V



## Plagiarism Checker X - Report

Originality Assessment

0%

Overall Similarity

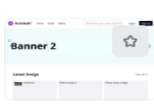
**Date:** Aug 3, 2025  
**Matches:** 0 / 49 words  
**Sources:** 0

**Remarks:** No similarity found,  
your document looks healthy.

**Verify Report:**  
Scan this QR Code



### D. Bukti Hosting Aplikasi/Website



**idcosmart**  
idcosmart.netlify.app  
Deploys from GitHub with Nuxt.  
Published on Jul 29.

Project configuration Add database

#### Analytics

Get a deeper understanding of your project with detailed analytics including page views, unique visitors, and top pages.

[Learn more](#)

Upgrade to enable

### E. Bukti Submit/Publish Artikel Ilmiah/HKI

Submit an Article

1. Start 2. Upload Submission 3. Enter Metadata 4. Confirmation 5. Next Steps

#### Submission complete

Thank you for your interest in publishing with JATSI (Jurnal Teknik Informatika dan Sistem Informasi).

#### What Happens Next?

The journal has been notified of your submission, and you've been emailed a confirmation for your records. Once the editor has reviewed the submission, they will contact you.

For now, you can:

- Review this submission
- Create a new submission
- Return to your dashboard