

# **PENERAPAN ARSITEKTUR CLIENT DATA LAYER MENGUNAKAN TANSTACK QUERY PADA DASHBOARD SENTIMENT ANALYSIS DENGAN METODE FOUNTAIN**

## **PROPOSAL TUGAS AKHIR**

Diajukan Sebagai Syarat Untuk Pengajuan Tugas Akhir Pada Program Studi Sarjana Terapan  
Teknologi Rekayasa Perangkat Lunak Jurusan Bisnis dan Informatika Politeknik Negeri  
Banyuwangi



Oleh:  
**NASRUL FAHMI ULUMUDDIN**  
**NIM 362258302204**

**PROGRAM STUDI SARJANA TERAPAN TEKNOLOGI REKAYASA  
PERANGKAT LUNAK  
JURUSAN BISNIS DAN INFORMATIKA  
POLITEKNIK NEGERI BANYUWANGI  
2026**

**PENERAPAN ARSITEKTUR CLIENT DATA LAYER  
MENGUNAKAN TANSTACK QUERY PADA DASHBOARD  
SENTIMENT ANALYSIS DENGAN METODE FOUNTAIN**

**PROPOSAL TUGAS AKHIR**

Diajukan kepada Jurusan Bisnis dan Informatika Politeknik Negeri Banyuwangi Sebagai  
Syarat Untuk Pengajuan Tugas Akhir Pada Program Studi Sarjana Terapan Teknologi  
Rekayasa Perangkat Lunak

Oleh:

**NASRUL FAHMI ULUMUDDIN**

**362258302204**

Pembimbing:

**Sepyan Purnama Kristanto, S.Kom., M.Kom**

**Arum Andary Ratri, S.Si., M.Si.**

**PROGRAM STUDI SARJANA TERAPAN TEKNOLOGI REKAYASA  
PERANGKAT LUNAK  
JURUSAN BISNIS DAN INFORMATIKA  
POLITEKNIK NEGERI BANYUWANGI  
2026**

# **LEMBAR PERSETUJUAN**

Proposal Tugas Akhir dengan Judul

## **PENERAPAN ARSITEKTUR CLIENT DATA LAYER MENGUNAKAN TANSTACK QUERY PADA DASHBOARD SENTIMENT ANALYSIS DENGAN METODE FOUNTAIN**

Disusun oleh:  
**NASRUL FAHMI ULUMUDDIN**  
**NIM 362258302204**

telah memenuhi syarat dan disetujui oleh Dosen Pembimbing untuk dilaksanakan Seminar  
Proposal Tugas Akhir bagi yang bersangkutan.

Banyuwangi, 5 Januari 2026

Mengetahui,  
Koordinator Program Studi,

Disetujui,  
Dosen Pembimbing 1,

Dianni Yusuf, S.Kom., M.Kom.  
NIP. NIPPPK.  
198403052021212004

Sepyan Purnama Kristanto, S.Kom., M.Kom  
NIP. NIP. 199009052019031024Nama  
Penguji 1

Banyuwangi, 5 Januari 2026

Disetujui,  
Dosen Pembimbing 2,

Arum Andary Ratri, S.Si., M.Si.  
NIP. NIP. 199209212020122021

# **Penerapan Arsitektur Client Data Layer Menggunakan TanStack Query pada Dashboard Sentiment Analysis dengan metode Fountain**

Oleh  
NASRUL FAHMI ULUMUDDIN  
NIM: 362258302204

## **ABSTRAK**

Abstrak adalah sebuah ringkasan singkat yang menjelaskan secara umum tentang isi dari laporan tugas akhir. Abstrak ditulis dalam tiga (3) paragraf yang berisi beberapa kalimat yang menyatakan tujuan, metode, hasil, dan kesimpulan dari laporan tugas akhir. Paragraf pertama berisi latar belakang dan tujuan tugas akhir. Paragraf kedua berisi metode dan pembahasannya. Paragraf ketiga berisi hasil dan simpulan dari tugas akhir yang dikerjakan.

Abstrak harus menjelaskan secara jelas dan singkat apa yang dibahas dalam laporan tugas akhir, mengapa penelitian ini penting dan apa yang ditemukan dari penelitian tersebut. Abstrak harus ditulis dengan bahasa yang mudah dipahami dan harus mencakup informasi penting yang dibahas dalam laporan tugas akhir.

Abstrak harus mengandung kata-kata yang relevan dengan laporan tugas akhir dan ditulis dengan bahasa yang formal dan akademik. Abstrak merupakan bagian penting dari sebuah laporan tugas akhir karena merupakan bagian yang pertama kali dibaca oleh pembaca dan harus dapat memberikan gambaran yang jelas tentang isi dari laporan tugas akhir. Oleh karena itu, abstrak harus ditulis dengan baik dan sebaik mungkin agar dapat memberikan gambaran yang jelas tentang laporan tugas akhir yang ditulis. Panjang abstrak sebaiknya dicukupkan dalam satu halaman, termasuk kata kunci. Tiga kata kunci dipandang cukup, yang masing-masingnya memuat paduan kata utama, yang dapat merepresentasikan isi Abstrak.

Kata kunci: Konsep Abstrak, Komponen Abstrak, Kata Kunci.

# **Implementation of Client Data Layer Architecture Using TanStack Query on Sentiment Analysis Dashboard with the Fountain Method**

by:

NASRUL FAHMI ULUMUDDIN

NIM: 362258302204

## **ABSTRACT**

The abstract is a short summary that explains in general the contents of the final assignment report. The abstract is written in three (3) paragraphs containing several sentences stating the objectives, methods, results and conclusions of the final assignment report. The first paragraph contains the background and objectives of the final assignment. The second paragraph contains the method and discussion. The third paragraph contains the results and conclusions of the final assignment carried out.

The abstract must explain clearly and concisely what is discussed in the final project report, why this research is important and what was found from the research. The abstract must be written in language that is easy to understand and must include important information discussed in the final project report.

The abstract must contain words that are relevant to the final project report and be written in formal and academic language. The abstract is an important part of a final assignment report because it is the part that is first read by the reader and must be able to provide a clear picture of the contents of the final assignment report. Therefore, the abstract must be written well and as well as possible in order to provide a clear picture of the final project report being written. The length of the abstract should be limited to one page, including keywords. Three keywords are considered sufficient, each of which contains a combination of main words, which can represent the contents of the Abstract.

Key words: Abstract Concepts, Abstract Components, Key Words.

## DAFTAR ISI

<b>HALAMAN SAMPUL</b> . . . . .	<b>i</b>
<b>LEMBAR PERSETUJUAN PROPOSAL</b> . . . . .	<b>ii</b>
<b>ABSTRAK</b> . . . . .	<b>iii</b>
<b>ABSTRACT</b> . . . . .	<b>iv</b>
<b>DAFTAR ISI</b> . . . . .	<b>v</b>
<b>DAFTAR SINGKATAN</b> . . . . .	<b>vii</b>
<b>DAFTAR GAMBAR</b> . . . . .	<b>viii</b>
<b>DAFTAR TABEL</b> . . . . .	<b>ix</b>
<b>BAB 1 PENDAHULUAN</b> . . . . .	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Perumusan Masalah . . . . .	3
1.3 Tujuan Proyek . . . . .	3
1.4 Manfaat Tugas Akhir . . . . .	3
1.5 Batasan Proyek . . . . .	4
<b>BAB 2 TINJAUAN PUSTAKA</b> . . . . .	<b>5</b>
2.1 Dasar Teori . . . . .	5
2.1.1.UMKM dan Digitalisasi . . . . .	5
2.1.2.Media Sosial sebagai Sumber Data . . . . .	6
2.1.3.Analisis Sentimen pada Media Sosial . . . . .	6
2.1.4.Dashboard Analitik dan Visualisasi Data . . . . .	7
2.1.5.Arsitektur Client Data Layer . . . . .	9
2.1.6.TanStack Query (React Query) . . . . .	12
2.1.7.REST API . . . . .	13
2.2 Penelitian Terkait . . . . .	14
<b>BAB 3 KONSEP RANCANGAN ALAT DAN PENGUJIAN</b> . . . . .	<b>15</b>
3.1 Metode Pengerjaan Project Berbasis Engineering Design Process . . . . .	15
3.1.1.Identifikasi Masalah . . . . .	15
3.1.2.Definisi Kebutuhan . . . . .	15
3.1.3.Generasi Ide dan Solusi . . . . .	15
3.1.4.Perencanaan dan Desain Awal . . . . .	15
3.1.5.Pembuatan Prototipe . . . . .	15
3.1.6.Pengujian dan Evaluasi . . . . .	16
3.1.7.Perbaikan dan Penyempurnaan . . . . .	16
3.2 Perancangan Sistem Elektronika . . . . .	16
3.2.1.Blok Diagram Sistem . . . . .	16
3.2.2.Pemilihan dan Spesifikasi Komponen Elektronika . . . . .	16
3.2.3.Perancangan Rangkaian Elektronika . . . . .	16
3.3 Perancangan Mekanik . . . . .	16
3.3.1.Spesifikasi Desain Mekanik . . . . .	16
3.3.2.Pemilihan Bahan dan Komponen Mekanik . . . . .	17
3.3.3.Desain Struktur dan Konstruksi . . . . .	17
3.4 Perancangan Perangkat Lunak . . . . .	17
3.4.1.Flowchart atau Diagram Alir Perangkat Lunak . . . . .	17
3.4.2.Pemrograman dan Pengembangan Kode . . . . .	17

3.4.	Pengujian Kode Perangkat Lunak . . . . .	17
3.5	Perancangan Integrasi Sistem . . . . .	17
3.5.	Integrasi Komponen Elektronika, Mekanik, dan Perangkat Lunak . . . . .	17
3.5.	Pengujian Awal dan Penyempurnaan Integrasi . . . . .	18
3.6	Rencana Pengujian . . . . .	18
3.6.	Metode Pengujian Sistem . . . . .	18
3.6.	Prosedur Pengujian . . . . .	18
3.6.	Kriteria Keberhasilan Pengujian . . . . .	18
<b>DAFTAR PUSTAKA . . . . .</b>		<b>19</b>

## DAFTAR SINGKATAN

FWHM	:	<i>Full width half maximum</i>
rms	:	<i>root mean square</i>
RFS	:	<i>Rotary forcespinning</i>
PVP	:	Polivinil pirolidon
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional



## **DAFTAR GAMBAR**

2.1	Architecture Client Data Layer .....	9
-----	--------------------------------------	---

## **DAFTAR TABEL**

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Usaha Mikro, Kecil, dan Menengah (UMKM) merupakan salah satu pilar utama dalam perekonomian Indonesia. Berdasarkan beberapa studi nasional, UMKM berkontribusi besar terhadap Produk Domestik Bruto (PDB) serta memiliki peran penting dalam penyerapan tenaga kerja. Namun, meskipun kontribusinya besar, tingkat literasi digital UMKM di Indonesia masih tergolong rendah, terutama dalam pemanfaatan teknologi informasi untuk analisis pasar dan pengambilan keputusan berbasis data. Penelitian oleh (Irianto et al., 2022) menyebutkan bahwa banyak UMKM belum mampu memanfaatkan teknologi digital secara optimal karena keterbatasan pengetahuan, akses, dan kemampuan mengelola data.

Di era digital saat ini, perilaku konsumen telah berubah secara signifikan. Media sosial seperti Instagram, TikTok, dan Facebook tidak hanya menjadi saluran pemasaran, tetapi juga menjadi ruang interaksi antara konsumen dan pelaku usaha. Studi oleh (Trulline, 2021) menunjukkan bahwa UMKM semakin bergantung pada media sosial untuk mempromosikan produk, membangun reputasi, dan memahami minat pasar. Komentar konsumen pada postingan media sosial mencerminkan persepsi publik terhadap produk atau layanan UMKM, sehingga dapat dimanfaatkan sebagai indikator kepuasan dan reputasi merek.

Namun, data komentar media sosial bersifat tidak terstruktur, jumlahnya besar, dan berubah secara dinamis, sehingga sulit dianalisis tanpa bantuan teknologi. Sejalan dengan (Joseph, 2024), Sentiment Analysis pada media sosial merupakan bidang penelitian yang berkembang pesat dan melibatkan penggunaan Natural Language Processing (NLP), text analysis, dan computational linguistics untuk mengidentifikasi serta mengekstraksi informasi subjektif dari data teks. Pendekatan NLP tersebut memungkinkan komentar tidak terstruktur diproses menjadi kategori sentimen yang dapat diinterpretasikan, baik positif, negatif, maupun netral. Dengan demikian, hasil analisis sentimen dari proses NLP dapat disajikan dalam bentuk dashboard interaktif agar UMKM dapat memahami tren sentimen, isu yang sering muncul, serta persepsi pelanggan secara lebih cepat dan intuitif.

Akan tetapi, pembangunan dashboard bukan hanya soal menampilkan data, melainkan bagaimana mengelola alur data di sisi frontend agar akurat, efisien, dan konsisten. Dashboard analitik biasanya memuat berbagai grafik, tabel, filter, dan indikator yang semuanya bergantung pada data API. Jika data dikelola secara tradisional misalnya dengan

mengonsumsi api langsung di setiap komponen React—akan muncul berbagai permasalahan teknis, antara lain: terjadinya permintaan API berulang, inkonsistensi data antar-komponen, waktu muat (loading time) yang lama, beban render tinggi karena komponen terus memperbarui data, sulit melakukan sinkronisasi state dan memelihara kode dalam jangka panjang.

Pada aplikasi data-driven seperti dashboard sentimen, masalah di atas dapat mengakibatkan pengalaman pengguna yang buruk dan data yang tidak akurat. Oleh karena itu, dibutuhkan pendekatan arsitektur modern yang mampu memisahkan logika pengelolaan data dari tampilan (UI), sekaligus memastikan setiap komponen menerima data yang konsisten, tersinkronisasi, dan tidak melakukan permintaan API berlebihan.

Salah satu solusi modern untuk tantangan tersebut adalah TanStack Query (sebelumnya React Query), sebuah pustaka server state management yang dirancang untuk menangani data asinkron dari API secara efisien. TanStack Query menyediakan fitur-fitur penting seperti caching otomatis, background refetching, query deduplication, stale-time management, prefetching, optimistic update, dan lazy-fetching berbasis kondisi melalui opsi enabled. Dengan mekanisme tersebut, frontend dapat mengelola data API seperti sebuah data layer yang mandiri, terstruktur, dan optimal.

Penelitian terbaru menunjukkan bahwa TanStack Query mampu mengurangi jumlah request API yang tidak diperlukan, meningkatkan performa aplikasi, serta menjaga konsistensi data antar-komponen sehingga sangat cocok diterapkan pada aplikasi berskala besar seperti dashboard analitik. Studi Comparing Performance of Redux, MobX, and React Query (Luz, 2025) menunjukkan bahwa React Query unggul dalam manajemen server state dibandingkan library state management lainnya, terutama dalam aplikasi yang memuat data dalam jumlah besar. Sementara itu, penelitian React Query and Lazy Loading Performance Optimization Best Practices (Micheal, 2025) menjelaskan bahwa kombinasi caching, lazy loading, dan query control dapat meningkatkan performa frontend secara signifikan.

Berdasarkan kebutuhan UMKM akan sistem pemantauan opini publik serta tantangan teknis dalam pengembangan dashboard analitik, penerapan arsitektur Client Data Layer menggunakan TanStack Query menjadi sangat relevan. Dengan pendekatan ini, pengembangan frontend dapat menjadi lebih efisien, terstruktur, dan mampu menampilkan data sentimen secara cepat, akurat, dan responsif. Oleh karena itu, penelitian ini berfokus pada penerapan arsitektur tersebut dalam pengembangan Dashboard Analisis Sentimen UMKM sebagai upaya mendukung digitalisasi UMKM dalam memahami persepsi konsumen dan meningkatkan daya saing.

## 1.2 Perumusan Masalah

1. Bagaimana menerapkan arsitektur Client Data Layer menggunakan TanStack Query pada studi kasus pengembangan Dashboard Analisis Sentimen UMKM?
2. Bagaimana penggunaan arsitektur Client Data Layer menggunakan TanStack Query dapat meningkatkan performa frontend?

## 1.3 Tujuan Proyek

1. Menerapkan arsitektur Client Data Layer berbasis TanStack Query pada sebuah kasus nyata, yaitu Dashboard Analisis Sentimen UMKM.
2. Mengevaluasi dampak penggunaan TanStack Query terhadap performa frontend pada studi kasus tersebut.
3. Menghasilkan dashboard analisis sentimen yang cepat, responsif, dan efektif digunakan pada konteks UMKM.

Dari tujuan tujuan tersebut, penelitian ini memiliki tujuan utama yaitu menerapkan arsitektur Client Data Layer berbasis TanStack Query pada sebuah kasus pengembangan Dashboard Analisis Sentimen UMKM, guna meningkatkan kualitas pengelolaan data pada sisi frontend. Penerapan arsitektur ini difokuskan pada pemanfaatan mekanisme caching, pengendalian data fetching, dan sinkronisasi data antar-komponen agar data yang ditampilkan lebih konsisten. Selain itu, penelitian ini bertujuan untuk mengevaluasi dampak penggunaan TanStack Query terhadap performa frontend dashboard, khususnya dari aspek waktu muat, jumlah permintaan API, dan responsivitas tampilan. Melalui penerapan tersebut, diharapkan dapat dihasilkan dashboard analisis sentimen yang cepat, responsif, serta efektif digunakan oleh UMKM dalam memahami persepsi konsumen berbasis data.

## 1.4 Manfaat Tugas Akhir

### Manfaat Teoritis

1. Penelitian ini diharapkan dapat menambah literatur mengenai penerapan TanStack Query dalam arsitektur data layer pada studi kasus aplikasi dashboard.

### Manfaat Praktis

#### 1. Manfaat bagi UMKM

Penelitian ini menghasilkan sebuah dashboard analisis sentimen yang dapat membantu UMKM dalam memahami komentar dan sentimen publik terhadap produk mereka secara lebih cepat. Dengan informasi tersebut, UMKM diharapkan memperoleh wawasan mengenai persepsi konsumen sebagai dasar pengambilan keputusan bisnis.

## **2. Manfaat bagi Pengembang**

Penelitian ini memberikan studi kasus penerapan arsitektur Client Data Layer menggunakan TanStack Query pada aplikasi React. Hasil penelitian ini dapat dijadikan acuan oleh pengembang frontend dalam merancang pengelolaan data yang lebih konsisten dan terstruktur.

## **3. Manfaat bagi Akademisi**

Penelitian ini menyediakan studi kasus yang dapat dijadikan referensi bagi akademisi maupun mahasiswa dalam penelitian sejenis, khususnya yang membahas arsitektur frontend, pengelolaan server state, dan pengembangan dashboard analitik.

### **1.5 Batasan Proyek**

Batasan proyek ditetapkan agar penelitian tetap terfokus pada tujuan yang telah dirumuskan. Adapun batasan proyek dalam Tugas Akhir ini adalah sebagai berikut:

1. Studi kasus yang digunakan adalah Dashboard Analisis Sentimen UMKM yang dikembangkan sebagai bagian dari proyek tim, dengan fokus penelitian pada perancangan dan implementasi arsitektur frontend.
2. Ruang lingkup penelitian hanya mencakup sisi frontend aplikasi, khususnya penerapan arsitektur Client Data Layer menggunakan TanStack Query dalam pengelolaan data API.
3. Proses analisis sentimen, termasuk pengumpulan data media sosial, preprocessing teks, dan penentuan kategori sentimen, tidak dibahas dalam penelitian ini. Data yang digunakan diperoleh melalui API dari backend dan extension browser.
4. Penelitian ini tidak mengevaluasi akurasi atau kualitas hasil analisis sentimen, melainkan hanya memanfaatkan hasil tersebut sebagai sumber data untuk dashboard.
5. Framework frontend yang digunakan adalah React dengan pustaka TanStack Query, tanpa melakukan perbandingan dengan framework atau library lain seperti Vue, Angular, Redux, atau MobX.
6. Evaluasi performa aplikasi hanya dilakukan pada konteks studi kasus dashboard yang dikembangkan, dengan fokus pada waktu muat data, jumlah permintaan API, mekanisme caching, dan konsistensi data antar-komponen.
7. Penelitian ini tidak membahas aspek optimasi backend, keamanan aplikasi, load testing, maupun pengujian usability secara mendalam

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Dasar Teori**

##### **2.1.1 UMKM dan Digitalisasi**

Usaha Mikro, Kecil, dan Menengah (UMKM) merupakan sektor usaha produktif yang memiliki peran penting dalam perekonomian, khususnya dalam menciptakan lapangan kerja dan mendorong pertumbuhan ekonomi lokal. UMKM umumnya memiliki karakteristik berupa skala usaha yang relatif kecil, keterbatasan modal, serta pengelolaan usaha yang masih sederhana. Dalam beberapa tahun terakhir, perkembangan teknologi digital telah mendorong UMKM untuk beradaptasi dengan perubahan lingkungan bisnis agar tetap mampu bersaing di tengah dinamika pasar yang semakin cepat.

Meskipun digitalisasi menawarkan berbagai peluang, UMKM masih menghadapi sejumlah tantangan dalam proses adopsinya. Tantangan tersebut meliputi rendahnya literasi digital, keterbatasan pemahaman dalam pemanfaatan teknologi informasi, serta kurangnya kemampuan dalam mengelola dan menganalisis data bisnis. Selain itu, salah satu kendala utama yang dihadapi UMKM adalah keterbatasan akses terhadap sumber daya yang dibutuhkan untuk mengembangkan usaha, seperti modal, informasi, dan teknologi (Alviani et al., 2025).

Permasalahan digitalisasi UMKM juga diperkuat oleh faktor demografis dan geografis. Menurut (Sofyan & Agusman, 2025), rendahnya literasi digital, khususnya pada pelaku UMKM usia lanjut dan yang berada di wilayah terpencil, menjadi hambatan dalam proses transformasi digital. Selain itu, tidak semua UMKM memiliki perangkat pendukung dan akses jaringan internet yang memadai, sehingga proses digitalisasi belum dapat diterapkan secara merata.

Dalam konteks pemasaran, pemanfaatan teknologi digital, khususnya media sosial, telah menjadi salah satu sarana utama bagi UMKM untuk mempromosikan produk dan menjangkau konsumen secara lebih luas. Aktivitas pemasaran digital tersebut menghasilkan data interaksi konsumen dalam jumlah besar yang berpotensi memberikan informasi berharga mengenai persepsi dan preferensi pasar. Oleh karena itu, UMKM membutuhkan sistem informasi yang mampu mengolah dan menyajikan data tersebut secara terstruktur dan informatif. Keberadaan sistem informasi berbasis dashboard analitik menjadi penting untuk mendukung pengambilan keputusan berbasis data serta meningkatkan efektivitas strategi pemasaran UMKM di era digital (Trulline, 2021).

### **2.1.2 Media Sosial sebagai Sumber Data**

Media sosial telah berkembang tidak hanya sebagai sarana komunikasi dan pemasaran, tetapi juga sebagai sumber data yang mencerminkan opini, persepsi, dan perilaku konsumen. Melalui media sosial, konsumen dapat mengekspresikan pengalaman serta keterlibatan mereka terhadap suatu merek melalui berbagai bentuk interaksi, seperti komentar, unggahan, tanda suka, dan aktivitas berbagi konten. Interaksi tersebut mencerminkan keterlibatan konsumen dan menghasilkan data yang bernilai untuk dianalisis lebih lanjut (Mardhatilah et al., 2024). Data yang dihasilkan dari media sosial umumnya bersifat tidak terstruktur dan terus bertambah secara dinamis, sehingga memerlukan sistem yang mampu mengelola dan menyajikan informasi tersebut secara terstruktur agar dapat dimanfaatkan secara optimal.

Data yang dihasilkan dari media sosial memiliki karakteristik bersifat tidak terstruktur, berjumlah besar, dan terus bertambah secara dinamis. Komentar dan ulasan konsumen umumnya berbentuk teks bebas yang mengandung opini subjektif, emosi, serta penilaian terhadap suatu produk atau layanan. Kondisi ini menyebabkan data media sosial sulit dianalisis secara manual. Oleh karena itu, diperlukan pendekatan sistematis untuk mengolah dan mengekstraksi informasi penting dari data tersebut agar dapat dimanfaatkan secara optimal.

Dalam konteks UMKM, data media sosial berpotensi memberikan wawasan penting terkait preferensi konsumen, tingkat kepuasan pelanggan, serta isu-isu yang sering muncul dalam interaksi publik. Informasi ini dapat dimanfaatkan sebagai dasar evaluasi strategi pemasaran dan pengambilan keputusan bisnis. Namun, tanpa dukungan sistem yang mampu mengelola dan menyajikan data secara terstruktur, potensi data media sosial tersebut sulit dimanfaatkan secara efektif.

Oleh karena itu, media sosial diposisikan sebagai salah satu sumber data utama dalam pengembangan sistem analitik bagi UMKM. Data yang diperoleh dari media sosial selanjutnya dapat diolah dan disajikan dalam bentuk informasi yang lebih ringkas dan mudah dipahami melalui dashboard analitik. Pendekatan ini memungkinkan UMKM untuk memantau persepsi konsumen dan dapat digunakan sebagai alat pengambilan keputusan.

### **2.1.3 Analisis Sentimen pada Media Sosial**

Analisis sentimen merupakan pendekatan analitik yang digunakan untuk mengidentifikasi dan mengklasifikasikan opini atau sikap pengguna terhadap suatu objek, seperti produk, layanan, atau merek, berdasarkan data teks. Dalam konteks media sosial, analisis sentimen memanfaatkan komentar, ulasan, dan berbagai bentuk interaksi pengguna untuk menentukan kecenderungan sentimen yang umumnya dikategorikan ke dalam sentimen



positif, negatif, atau netral. Pendekatan ini relevan karena media sosial menyediakan data opini konsumen yang bersifat spontan, terbuka, dan dihasilkan secara terus-menerus melalui interaksi pengguna.

Seiring dengan meningkatnya aktivitas pengguna di media sosial, volume data opini yang dihasilkan juga semakin besar dan bersifat dinamis. Data tersebut umumnya tidak terstruktur dan mengandung unsur subjektivitas, emosi, serta bahasa informal, sehingga sulit dianalisis secara manual. Oleh karena itu, analisis sentimen digunakan untuk menyederhanakan data teks yang kompleks menjadi informasi yang lebih ringkas dan terstruktur, sehingga dapat digunakan untuk memahami kecenderungan opini publik dan perilaku konsumen secara lebih sistematis.

Penelitian yang dilakukan oleh (Fajarini et al., 2025) menunjukkan bahwa analisis sentimen berbasis data media sosial merupakan metode yang inovatif dan efektif dalam memprediksi tren pasar serta memahami perilaku konsumen. Dengan memanfaatkan data berskala besar yang dihasilkan secara real time oleh pengguna media sosial, analisis sentimen mampu mengenali pola opini publik, preferensi konsumen, serta perubahan tren yang relevan di berbagai sektor industri. Hasil penelitian tersebut juga menegaskan bahwa informasi yang dihasilkan dari analisis sentimen dapat memberikan wawasan yang bernilai untuk mendukung pengambilan keputusan bisnis berbasis data.

Lebih lanjut, (Fajarini et al., 2025) menyatakan bahwa analisis sentimen berbasis media sosial menawarkan pendekatan yang lebih cepat dan efisien dibandingkan metode konvensional, seperti survei manual atau riset pasar tradisional, karena memungkinkan pemantauan opini publik secara berkelanjutan. Dengan demikian, analisis sentimen dipandang sebagai alat yang penting dalam mendukung strategi bisnis modern yang berbasis data dan responsif terhadap dinamika pasar.

Dalam penelitian ini, analisis sentimen diposisikan sebagai proses pengolahan data yang dilakukan pada sisi backend. Fokus penelitian tidak terletak pada metode atau algoritma analisis sentimen yang digunakan, melainkan pada pemanfaatan hasil analisis sentimen sebagai sumber data yang dikelola dan disajikan melalui dashboard analitik di sisi frontend. Hasil analisis sentimen tersebut digunakan sebagai input utama untuk mendukung penyajian informasi yang informatif, terstruktur, dan mudah dipahami oleh pengguna.

#### **2.1.4 Dashboard Analitik dan Visualisasi Data**

Dashboard analitik merupakan sistem informasi yang dirancang untuk menyajikan data dalam bentuk visual yang ringkas, terintegrasi, dan mudah dipahami oleh pengguna. Dashboard

ini umumnya memanfaatkan berbagai komponen visualisasi, seperti grafik, tabel, dan indikator kinerja, untuk menampilkan informasi penting yang mendukung proses pemantauan, analisis, dan pengambilan keputusan. Berbeda dengan laporan statis, dashboard analitik bersifat dinamis dan interaktif, sehingga memungkinkan pengguna untuk memperoleh gambaran kondisi secara cepat dan menyeluruh.

Visualisasi data memiliki peran penting dalam dashboard analitik karena mampu mengubah data mentah menjadi informasi yang lebih bermakna dan intuitif. Melalui visualisasi yang tepat, pengguna dapat dengan mudah mengidentifikasi pola, tren, serta perubahan yang terjadi pada data. Visualisasi data juga berfungsi sebagai sarana penyampaian informasi yang bersifat naratif, di mana kinerja dan kondisi bisnis dapat digambarkan secara komprehensif tanpa harus melalui proses analisis data yang kompleks.

Penelitian yang dilakukan oleh (Rathore et al., 2025) menunjukkan bahwa penerapan teknik visualisasi data yang efektif dapat membantu pengambil keputusan dalam menghasilkan keputusan yang lebih informatif, akurat, dan ringkas. Studi tersebut menegaskan bahwa dashboard, berbagai jenis grafik, serta pendekatan visualisasi yang terstruktur berperan penting dalam meningkatkan kualitas business intelligence dan mendukung perencanaan strategis. Selain itu, visualisasi data dinilai mampu meningkatkan efisiensi proses pengambilan keputusan dan mengurangi waktu yang dibutuhkan untuk menganalisis data, sehingga organisasi dapat merespons permasalahan dan dinamika bisnis secara lebih cepat.

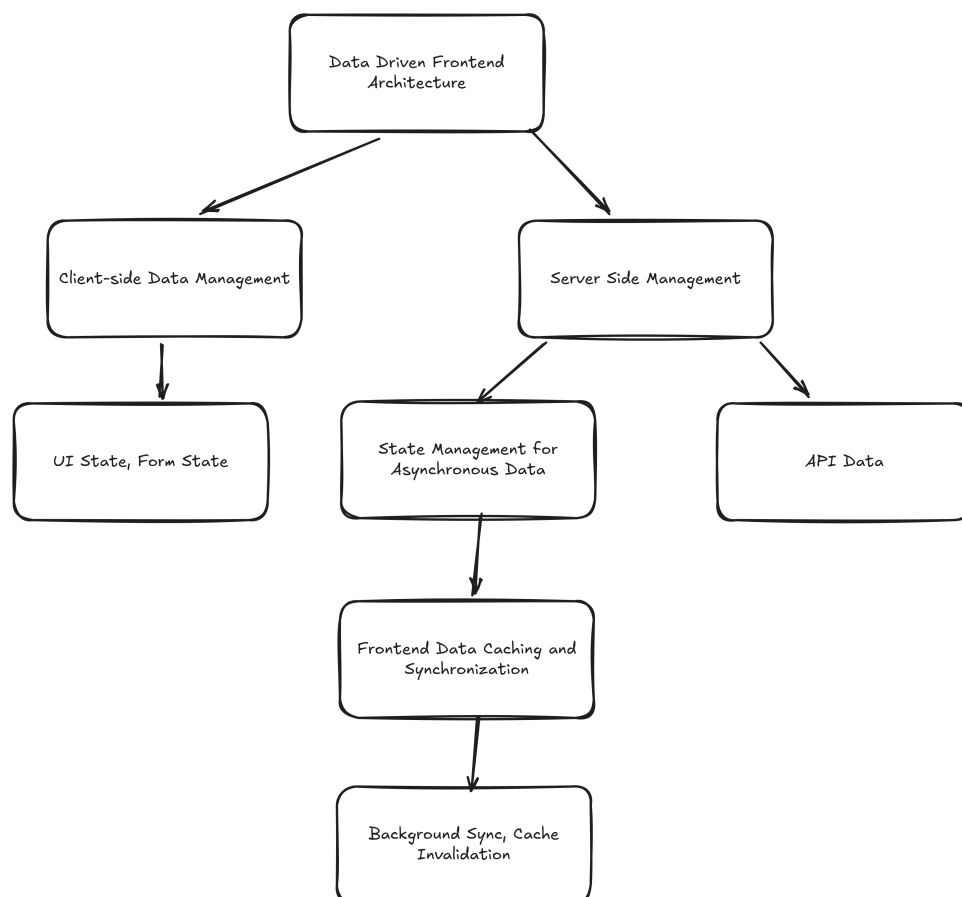
Dalam konteks aplikasi data-driven, dashboard analitik berfungsi sebagai jembatan antara data dan pengambilan keputusan. Dashboard memungkinkan penyajian data secara real time atau near real time, sehingga perubahan kondisi dan tren dapat dipantau secara berkelanjutan. Hal ini menjadikan dashboard analitik sebagai komponen penting dalam sistem yang memanfaatkan data berskala besar dan bersifat dinamis, termasuk data yang dihasilkan dari media sosial.

Bagi UMKM, keberadaan dashboard analitik menjadi sangat relevan karena dapat menyederhanakan informasi yang kompleks menjadi tampilan visual yang mudah dipahami. Informasi seperti kecenderungan sentimen konsumen, pola interaksi pengguna, dan ringkasan data pemasaran dapat disajikan secara visual, sehingga pelaku UMKM tidak perlu melakukan analisis data secara manual. Dengan demikian, dashboard analitik dapat mendukung UMKM dalam mengambil keputusan bisnis yang lebih cepat, tepat, dan berbasis data.

### 2.1.5 Arsitektur Client Data Layer

Pada aplikasi frontend modern yang bersifat data-driven, pengelolaan data yang bersumber dari Application Programming Interface (API) menjadi salah satu aspek penting dalam arsitektur sistem. Aplikasi seperti dashboard analitik umumnya menampilkan berbagai komponen antarmuka, seperti grafik, tabel, dan indikator, yang bergantung pada data yang sama dan diperbarui secara berkala. Jika pengambilan dan pengolahan data dilakukan secara langsung di setiap komponen antarmuka, maka dapat menimbulkan berbagai permasalahan, seperti permintaan API yang berulang, inkonsistensi data antar-komponen, serta peningkatan beban render yang berdampak pada penurunan performa aplikasi.

Client Data Layer merupakan pendekatan arsitektural dalam pengembangan frontend modern yang bertujuan untuk mengelola data yang bersumber dari server secara terpusat di sisi klien. Pendekatan ini muncul sebagai respons terhadap meningkatnya kompleksitas aplikasi data-driven, dimana data bersifat asinkron, dinamis, dan digunakan oleh banyak komponen antarmuka secara bersamaan. Dengan adanya Client Data Layer, proses fetching, caching, dan synchronize data dapat dilakukan secara lebih terstruktur, sehingga membantu menjaga konsistensi data dan meningkatkan efisiensi aplikasi.



**Gambar 2.1** Architecture Client Data Layer

## 1. Server State Management

Server state management mengacu pada pengelolaan data yang bersumber dari sistem eksternal, seperti API atau layanan backend, yang bersifat asinkron dan dapat berubah di luar kendali langsung aplikasi klien. Data ini memiliki karakteristik dinamis karena dipengaruhi oleh kondisi jaringan, waktu respons server, serta pembaruan data di sisi backend. Oleh karena itu, server state memerlukan mekanisme khusus untuk menangani proses pengambilan data, status pemuatan, penanganan kesalahan, serta pembaruan dan sinkronisasi data agar informasi yang digunakan oleh berbagai komponen antarmuka tetap konsisten dan akurat.

## 2. Client-side Data Management

Client-side data management berfokus pada pengelolaan data di sisi klien setelah data tersebut diperoleh dari server, termasuk penyimpanan sementara, penggunaan ulang data, serta distribusi data ke berbagai komponen antarmuka. Pendekatan ini bertujuan untuk mengurangi ketergantungan terhadap permintaan data berulang ke server, sehingga dapat meningkatkan efisiensi aplikasi, mempercepat waktu muat, dan menjaga konsistensi informasi yang ditampilkan. Dengan pengelolaan data yang terstruktur di sisi klien, aplikasi frontend dapat menyajikan data secara lebih responsif dan stabil, khususnya pada aplikasi yang bersifat data-driven seperti dashboard analitik.

## 3. Data-driven Frontend Architecture

Data-driven Frontend Architecture merupakan pendekatan pengembangan di mana seluruh antarmuka pengguna didorong oleh data sebagai sumber kebenaran utama. Dalam pola ini, UI dihasilkan sebagai fungsi dari state yang ada—artinya, setiap perubahan pada data akan secara otomatis memicu pembaruan pada tampilan aplikasi. Arsitektur ini sangat berguna untuk aplikasi yang menampilkan informasi dinamis seperti dashboard analitik, papan monitoring, atau aplikasi dengan data real-time. Keunggulan utamanya adalah konsistensi tampilan yang lebih terjamin, alur data yang mudah dilacak, dan pengembangan fitur baru yang lebih sistematis karena UI dan logika data terpisah dengan jelas.

## 4. State Management for Asynchronous Data

State Management for Asynchronous Data adalah pendekatan khusus untuk menangani data yang diperoleh melalui state, seperti panggilan API, operasi file, atau permintaan jaringan lainnya. Karena data tersebut tidak tersedia secara instan,

sistem harus mampu mengelola berbagai state yang mungkin terjadi: mulai dari state (idle), state fetching atau onloading, state success, hingga state error. Tantangan utamanya adalah memastikan aplikasi tetap responsif dan memberikan umpan balik yang informatif kepada pengguna selama proses pengambilan data. Pendekatan ini juga mencakup strategi seperti pembatalan permintaan yang tidak diperlukan, pengulangan otomatis saat gagal, dan pembaruan data latar belakang untuk menjaga informasi tetap konsisten.

#### 5. Frontend Data Caching and Synchronization

Data Caching and Synchronization adalah teknik untuk meningkatkan kinerja aplikasi dengan menyimpan salinan data dari server di memori klien. Dengan adanya cache, aplikasi dapat menampilkan informasi secara cepat tanpa perlu melakukan permintaan berulang ke server. Namun, teknik ini juga menimbulkan tantangan, yaitu data yang disimpan dapat menjadi kedaluwarsa jika terjadi perubahan di sisi server. Oleh karena itu, diperlukan mekanisme sinkronisasi yang cerdas, seperti pembaruan di latar belakang (background refresh) dan penandaan cache yang kedaluwarsa (cache invalidation). Dengan pendekatan ini, aplikasi dapat mencapai kecepatan optimal sekaligus memastikan keakuratan data yang ditampilkan.

Beberapa penelitian menunjukkan bahwa penerapan mekanisme caching pada aplikasi frontend dapat meningkatkan performa dan responsivitas sistem. Caching memungkinkan data hasil pemanggilan API disimpan sementara di sisi klien sehingga dapat digunakan kembali tanpa melakukan permintaan ulang ke server. Pendekatan ini terbukti mampu mengurangi duplikasi permintaan API dan mempercepat waktu respons aplikasi, terutama pada aplikasi berbasis framework JavaScript yang bersifat data-driven. Pemanfaatan pustaka seperti TanStack Query dalam mengelola caching dan prefetching data juga dinilai efektif dalam meningkatkan efisiensi pengelolaan data di sisi frontend serta pengalaman pengguna secara keseluruhan (Rahman & Prihanto, 2024).

Penerapan Client Data Layer memberikan sejumlah manfaat dalam pengembangan aplikasi frontend. Salah satu manfaat utama adalah peningkatan konsistensi data, di mana beberapa komponen yang membutuhkan data yang sama dapat memperoleh informasi yang seragam tanpa harus melakukan permintaan data secara terpisah. Selain itu, Client Data Layer memungkinkan pengurangan jumlah permintaan API yang tidak diperlukan melalui mekanisme caching dan pengelolaan siklus data. Pendekatan ini juga mendukung efisiensi

waktu muat aplikasi serta mempermudah pengelolaan data yang bersifat asinkron dan dinamis.

Dalam konteks dashboard analitik, keberadaan Client Data Layer menjadi semakin penting karena data yang ditampilkan umumnya bersifat besar, sering diperbarui, dan digunakan oleh banyak komponen secara bersamaan. Dengan memanfaatkan Client Data Layer, dashboard dapat menampilkan data secara lebih responsif dan stabil, sekaligus meminimalkan risiko inkonsistensi informasi yang ditampilkan kepada pengguna. Pendekatan ini mendukung terciptanya arsitektur frontend yang lebih terorganisasi, mudah dipelihara, dan skalabel.

#### **2.1.6 TanStack Query (React Query)**

TanStack Query merupakan pustaka manajemen data pada sisi frontend yang dirancang untuk mengelola data yang bersumber dari server (server state) secara efisien. Dalam dokumentasi resminya, TanStack Query dijelaskan sebagai “the missing data-fetching layer for web applications” yang berfungsi untuk mempermudah proses pengambilan, penyimpanan sementara (caching), sinkronisasi, serta pembaruan data dari server (Tanstack LCC, 2025). Pendekatan ini ditujukan untuk menangani kompleksitas data asinkron yang tidak dapat dikelola secara optimal menggunakan mekanisme state management konvensional.

Dalam dokumentasi resminya, TanStack Query mendefinisikan server state sebagai data yang berasal dari sumber eksternal dan memiliki karakteristik asinkron, dapat berubah sewaktu-waktu, serta memerlukan mekanisme khusus untuk menjaga konsistensi data di sisi klien. Oleh karena itu, TanStack Query menyediakan pendekatan deklaratif dalam pengelolaan server state, di mana pengembang dapat mendefinisikan kebutuhan data tanpa harus menangani secara manual proses sinkronisasi dan pembaruan data di setiap komponen antarmuka (TanStack Documentation, 2024).

Salah satu fitur utama TanStack Query adalah mekanisme caching yang memungkinkan data hasil pemanggilan API disimpan sementara di sisi klien. Dengan adanya caching, data yang telah diperoleh dapat digunakan kembali oleh komponen lain tanpa perlu melakukan permintaan ulang ke server, selama data tersebut masih dianggap valid. Pendekatan ini berkontribusi dalam mengurangi jumlah permintaan API yang tidak diperlukan, meningkatkan efisiensi aplikasi, serta mempercepat waktu respons antarmuka pengguna.

Selain caching, TanStack Query juga menyediakan mekanisme sinkronisasi data yang mendukung pembaruan data secara otomatis. Melalui konsep seperti refetching dan invalidasi data, TanStack Query memastikan bahwa data yang ditampilkan tetap mutakhir ketika terjadi perubahan di sisi server. Mekanisme ini sangat relevan pada aplikasi data-driven, seperti

dashboard analitik, yang menampilkan data secara dinamis dan digunakan oleh banyak komponen secara bersamaan.

Dalam konteks arsitektur frontend, TanStack Query dapat diposisikan sebagai implementasi konkret dari Client Data Layer. Pustaka ini berperan sebagai lapisan perantara antara backend API dan komponen antarmuka pengguna, sehingga komponen UI tidak berinteraksi langsung dengan API. Dengan demikian, TanStack Query membantu memisahkan logika pengelolaan data dari logika tampilan, meningkatkan keterbacaan kode, serta mempermudah pemeliharaan aplikasi dalam jangka panjang.

Pada penelitian ini, TanStack Query digunakan sebagai solusi untuk menerapkan arsitektur Client Data Layer pada pengembangan dashboard analitik. Fokus penggunaan TanStack Query diarahkan pada pengelolaan server state, caching data, serta sinkronisasi data antar-komponen, tanpa membahas aspek internal pustaka atau detail implementasi secara mendalam. Dengan pendekatan ini, TanStack Query berperan sebagai fondasi pengelolaan data pada sisi frontend yang mendukung penyajian informasi sentimen secara konsisten, responsif, dan efisien.

### **2.1.7 REST API**

Representational State Transfer Application Programming Interface (REST API) merupakan gaya arsitektur layanan web yang digunakan untuk memungkinkan komunikasi antara klien dan server melalui protokol HTTP secara terstandarisasi. REST API bersifat stateless, di mana setiap permintaan dari klien harus membawa seluruh informasi yang dibutuhkan untuk diproses oleh server, sehingga tidak bergantung pada status permintaan sebelumnya. Data yang dipertukarkan umumnya disajikan dalam format JSON karena bersifat ringan dan mudah diproses oleh aplikasi frontend, menjadikan REST API banyak digunakan pada aplikasi web modern yang bersifat data-driven.

Dalam konteks aplikasi frontend analitik, REST API berperan sebagai sumber utama data (server state) yang dikonsumsi oleh antarmuka pengguna. Data yang diperoleh melalui REST API bersifat asinkron dan dapat berubah sewaktu-waktu, sehingga memerlukan mekanisme pengelolaan data yang mampu menangani proses pengambilan, pembaruan, dan sinkronisasi data secara efisien. Penelitian terkini menunjukkan bahwa REST API memiliki keunggulan dalam penyajian data yang bersifat datar dan mudah di-cache, sehingga mampu meningkatkan efisiensi distribusi data serta memperbesar rasio cache hit pada lapisan jaringan. Pendekatan ini dinilai lebih optimal untuk kebutuhan aplikasi yang menampilkan data terstruktur secara berulang, seperti dashboard dan sistem pelaporan, dibandingkan pendekatan

API lain yang lebih kompleks (Islam, 2025).

Lebih lanjut, penelitian tersebut menegaskan bahwa performa aplikasi web tidak ditentukan oleh satu teknologi tertentu, melainkan oleh keselarasan antara desain akses data, mekanisme caching, serta pengelolaan state pada sisi klien. REST API yang dirancang dengan kontrak data yang jelas dan cache-aware terbukti mendukung peningkatan performa dan skalabilitas aplikasi ketika dipadukan dengan lapisan pengelolaan data di frontend. Oleh karena itu, dalam pengembangan aplikasi frontend modern, REST API umumnya tidak diakses secara langsung oleh setiap komponen antarmuka, melainkan melalui lapisan pengelolaan data seperti Client Data Layer agar data dapat dikelola secara terpusat, konsisten, dan efisien.

Dalam penelitian ini, REST API diposisikan sebagai penyedia data hasil analisis sentimen yang diproses di sisi backend. Data tersebut selanjutnya dikelola pada sisi frontend melalui arsitektur Client Data Layer sebelum ditampilkan dalam bentuk visualisasi pada dashboard analitik. Dengan pemisahan peran ini, REST API berfungsi sebagai sumber data, sementara pengelolaan performa, caching, dan sinkronisasi data dilakukan sepenuhnya di sisi frontend untuk mendukung penyajian informasi yang responsif dan konsisten.

## **2.2 Penelitian Terkait**



## **BAB 3**

### **KONSEP RANCANGAN ALAT DAN PENGUJIAN**

#### **3.1 Metode Pengerjaan Project Berbasis Engineering Design Process**

Bagian ini menjelaskan metode pengerjaan proyek yang mengadopsi pendekatan Engineering Design Process. Pendekatan ini digunakan untuk memastikan perancangan dan pengembangan proyek dilakukan secara sistematis, mulai dari identifikasi masalah hingga evaluasi akhir. Langkah-langkah dalam metode ini membantu dalam mencapai solusi yang optimal dan terukur.

##### **3.1.1 Identifikasi Masalah**

Tahap identifikasi masalah bertujuan untuk menguraikan permasalahan utama yang dihadapi dan memerlukan solusi. Pada tahap ini, dilakukan analisis untuk memahami aspek-aspek penting dari masalah dan menentukan faktor-faktor yang perlu diatasi melalui proyek ini.

##### **3.1.2 Definisi Kebutuhan**

Berdasarkan masalah yang telah diidentifikasi, tahap ini berfokus pada definisi kebutuhan proyek secara jelas dan terstruktur. Kebutuhan ini meliputi spesifikasi teknis, fungsi yang diinginkan, serta kriteria-kriteria lain yang harus dipenuhi agar solusi dapat berfungsi dengan baik.

##### **3.1.3 Generasi Ide dan Solusi**

Pada tahap ini, berbagai ide dan solusi alternatif dikembangkan dan dievaluasi. Bagian ini menjelaskan proses brainstorming untuk menghasilkan ide yang inovatif, termasuk analisis terhadap kelebihan dan kekurangan dari setiap alternatif solusi yang diusulkan.

##### **3.1.4 Perencanaan dan Desain Awal**

Desain awal sistem dikembangkan berdasarkan solusi yang dipilih, dengan mempertimbangkan aspek teknis dan kebutuhan yang telah didefinisikan. Bagian ini menyajikan perencanaan mengenai struktur sistem, alat, dan bahan yang akan digunakan, serta jadwal kerja proyek secara keseluruhan.

##### **3.1.5 Pembuatan Prototipe**

Prototipe dibuat untuk menguji konsep dan desain awal dari sistem. Bagian ini menguraikan langkah-langkah dalam proses pembuatan prototipe, termasuk alat dan bahan

yang diperlukan, serta tantangan yang mungkin dihadapi selama proses.

### **3.1.6 Pengujian dan Evaluasi**

Prototipe yang telah dibuat kemudian diuji untuk menilai kinerjanya terhadap kebutuhan dan spesifikasi yang telah ditetapkan. Bagian ini menjelaskan prosedur pengujian yang diterapkan, metode pengumpulan data, serta analisis terhadap hasil yang diperoleh.

### **3.1.7 Perbaikan dan Penyempurnaan**

Berdasarkan hasil pengujian dan evaluasi, dilakukan perbaikan untuk menyempurnakan sistem. Bagian ini menjelaskan penyesuaian yang dilakukan untuk meningkatkan kinerja sistem, serta proses iterasi yang dilakukan hingga mencapai hasil yang optimal.

## **3.2 Perancangan Sistem Elektronika**

Bagian ini menguraikan perancangan dari sisi elektronika yang menjadi inti dari sistem. Perancangan ini mencakup blok diagram, pemilihan komponen, dan perancangan rangkaian.

### **3.2.1 Blok Diagram Sistem**

Blok diagram sistem memberikan gambaran umum mengenai arsitektur sistem secara keseluruhan. Bagian ini menyajikan diagram beserta penjelasan fungsi setiap blok yang terdapat di dalam sistem, termasuk bagaimana setiap blok berinteraksi.

### **3.2.2 Pemilihan dan Spesifikasi Komponen Elektronika**

Pemilihan komponen elektronika dilakukan berdasarkan kebutuhan dari desain sistem. Bagian ini menjelaskan spesifikasi teknis dari setiap komponen yang digunakan, seperti mikrokontroler, sensor, aktuator, dan komponen pendukung lainnya.

### **3.2.3 Perancangan Rangkaian Elektronika**

Perancangan rangkaian elektronika bertujuan untuk mencapai fungsionalitas yang diinginkan dari sistem. Bagian ini menguraikan skema rangkaian, penjelasan aliran arus dan tegangan, serta hubungan antar komponen dalam sistem.

## **3.3 Perancangan Mekanik**

Bagian ini membahas perancangan mekanik dari sistem yang mendukung komponen elektronik secara fisik, termasuk struktur dan pemilihan bahan.

### **3.3.1 Spesifikasi Desain Mekanik**

Desain mekanik disusun berdasarkan kebutuhan fisik sistem untuk memastikan bahwa komponen elektronika terlindungi dan dapat berfungsi dengan baik. Bagian ini menjelaskan

spesifikasi teknis desain mekanik yang digunakan.

### **3.3.2 Pemilihan Bahan dan Komponen Mekanik**

Pemilihan bahan didasarkan pada kriteria seperti kekuatan, ketahanan, dan biaya. Bagian ini menguraikan bahan dan komponen mekanik yang digunakan untuk konstruksi sistem, serta alasan pemilihan bahan tersebut.

### **3.3.3 Desain Struktur dan Konstruksi**

Struktur dan konstruksi sistem dirancang untuk memberikan dukungan fisik yang stabil. Bagian ini menjelaskan proses desain dan konstruksi dari struktur mekanik sistem, serta tantangan yang mungkin dihadapi.

## **3.4 Perancangan Perangkat Lunak**

Bagian ini mencakup perancangan perangkat lunak yang mengendalikan sistem, meliputi diagram alir, pengembangan kode, dan pengujian perangkat lunak.

### **3.4.1 Flowchart atau Diagram Alir Perangkat Lunak**

Diagram alir menggambarkan alur kerja dari perangkat lunak yang mengontrol sistem. Bagian ini menyajikan flowchart lengkap yang menunjukkan logika dan struktur kontrol dari perangkat lunak.

### **3.4.2 Pemrograman dan Pengembangan Kode**

Kode perangkat lunak dikembangkan untuk mendukung fungsionalitas sistem. Bagian ini menguraikan struktur dan logika dari program yang dibuat, bahasa pemrograman yang digunakan, serta strategi pengembangan kode.

### **3.4.3 Pengujian Kode Perangkat Lunak**

Setelah kode perangkat lunak dikembangkan, dilakukan pengujian untuk memastikan bahwa perangkat lunak berfungsi sesuai yang diharapkan. Bagian ini menjelaskan metode pengujian, jenis uji (misalnya, uji unit dan uji integrasi), serta hasil yang diperoleh.

## **3.5 Perancangan Integrasi Sistem**

Bagian ini menjelaskan proses integrasi antara komponen elektronik, mekanik, dan perangkat lunak agar sistem dapat bekerja sebagai satu kesatuan.

### **3.5.1 Integrasi Komponen Elektronika, Mekanik, dan Perangkat Lunak**

Proses integrasi bertujuan untuk menyatukan seluruh komponen sistem agar berfungsi sebagai satu kesatuan. Bagian ini menguraikan langkah-langkah dalam proses integrasi dan

memastikan semua komponen bekerja secara sinkron.

### **3.5.2 Pengujian Awal dan Penyempurnaan Integrasi**

Pengujian awal dilakukan setelah integrasi untuk menilai performa sistem secara keseluruhan. Bagian ini menjelaskan hasil pengujian integrasi dan modifikasi yang diperlukan untuk meningkatkan kinerja sistem.

## **3.6 Rencana Pengujian**

Bagian ini merencanakan pengujian yang komprehensif terhadap sistem untuk memastikan bahwa semua komponen dan fungsi bekerja dengan baik.

### **3.6.1 Metode Pengujian Sistem**

Metode pengujian dipilih berdasarkan tujuan dan kebutuhan proyek. Bagian ini menjelaskan berbagai metodologi pengujian yang dirancang, seperti uji kinerja, uji stabilitas, dan uji kompatibilitas.

#### **3.6.2 Prosedur Pengujian**

Prosedur pengujian disusun untuk menguji sistem secara menyeluruh, mulai dari pengaturan awal hingga pelaksanaan uji. Bagian ini menjelaskan langkah-langkah pengujian secara detail untuk menjamin konsistensi dan keandalan hasil.

#### **3.6.3 Kriteria Keberhasilan Pengujian**

Kriteria keberhasilan ditentukan untuk mengevaluasi kinerja sistem berdasarkan parameter-parameter tertentu. Bagian ini menguraikan kriteria-kriteria keberhasilan yang digunakan, seperti ketepatan, keandalan, dan efisiensi sistem.

Pada bagian ini akan dijelaskan beberapa tahap utama dalam proses pengembangan proyek dan beberapa rekomendasi peningkatan yang dapat dilakukan. Daftar tahapan pengembangan proyek akan disajikan dalam bentuk bernomor untuk menunjukkan urutan logis dari proses, sementara rekomendasi peningkatan akan disajikan dalam bentuk daftar berpoin untuk mempermudah identifikasi setiap item secara mandiri.

## DAFTAR PUSTAKA

- Alviani, N. A., Studi, P., Fakultas, M., & Bangsa, U. B. (2025). Transformasi Digital pada UMKM dalam Meningkatkan Daya Saing Pasar. *Master Manajemen*, 3(1), 134–140.
- Fajarini, S., Kurniawati, J., & Yuliani, F. (2025). Social media sentiment analysis as a new tool for predicting market trends and consumer behaviour. *Proceeding of International Conference on Social Science and Humanity*, 2, 899–909.
- Irianto, H., Viesta, A. D., Nugroho, A. T., Wahyuni, T., Prabowo, W. C., Hamid, I. N., Anufah, T. N., Permatasari, H. I., Salsabila, A., Sofyana, & Hardiyanti, F. Y. (2022). Digitalisasi UMKM sebagai Upaya Peningkatan Pemasaran dan Penjualan Online di Desa Tengkluk. *Journal of Cooperative, Small and Medium Enterprise Development*, 1(2), 60–64.
- Islam, M. M. (2025). The impact of data-driven web frameworks on performance and scalability of u.s. enterprise applications. *International Journal of Business and Economics Insights*, 05, 523–558.
- Joseph, T. (2024). Natural Language Processing (NLP) for Sentiment Analysis in Social Media. *International Journal of Computing and Engineering*, 6(2), 35–48.
- Luz, H. (2025). Comparing Performance of Redux, MobX, and React Query. Preprint, ResearchGate.
- Mardhatilah, D., Omar, A., & Septiari, E. D. (2024). BUILDING CONSUMER ENGAGEMENT IN SOCIAL MEDIA: A SYSTEMATIC LITERATURE REVIEW. *JOURNAL OF BUSINESS MANAGEMENT AND ACCOUNTING*, 14(1), 1–35.
- Micheal, D. (2025). React Query and Lazy Loading: Performance Optimization Best Practices. Preprint, ResearchGate.
- Rahman, A. & Prihanto, A. (2024). Optimisasi Kinerja Aplikasi Fitness Berbasis Next.js Melalui Penerapan Metode Caching Pada PT. Anugerah Wijaya Raga. *Journal of Informatics and Computer Science (JINACS)*, 6(02), 333–340.
- Rathore, S., Nawkhare, R., Sharma, N., Chaudhary, N., Chakole, S., & Vishwakrama, B. (2025). Effective data visualization techniques for business decision-makers. *International Research Journal on Advanced Engineering and Management (IRJAEM)*, 3, 2029–2036.
- Sofyan, S. & Agusman (2025). ANALISIS KESIAPAN UMKM MENGHADAPI EKONOMI DIGITAL. *Smart Jurnal Ilmiah*, IX(1), 1–4.
- Tanstack LCC (2025). Tanstack Query Official Documentation.
- Trulline, P. (2021). Pemasaran produk UMKM melalui media sosial dan e-commerce. *Jurnal Manajemen Komunikasi*, 5(2), 259.