

AI Market Pulse Radar

Complete Implementation Guide v2.0

Target: 220/220 Poin (100%)

IMPHNEN x Kolosal.ai Hackathon 2025

Tim Hackathon - 5 Orang, 7 Hari

DOKUMENINI BERISI

1. Konsep aplikasi yang JELAS (Dashboard, BUKAN Chatbot!)
2. Strategi Authentication lengkap (Demo Mode vs Full Auth)
3. Database Schema detail dengan multi-user support
4. Dataset requirements dan cara generate sample data
5. User Flow interaksi yang detail
6. Analisis LENGKAP rubrik penilaian
7. Formula matematika dengan PENJELASAN LENGKAP (tanpa code!)
8. Pembagian tugas SPESIFIK per orang per hari
9. Template README yang SIAP PAKAI
10. Script video demo LENGKAP
11. Checklist ANTI-GAGAL sebelum submit

Contents

I KONSEP DAN STRATEGI	2
1 Konsep Aplikasi: Dashboard Analytics, BUKAN Chatbot	2
1.1 Apa itu AI Market Pulse Radar?	2
1.2 Kenapa BUKAN Chatbot?	2
1.3 Perbandingan: Chatbot vs Dashboard	3
1.4 User Flow: Bagaimana User Berinteraksi?	3
1.5 Flow Lengkap dari Awal sampai Akhir	3
2 Authentication Strategy	5
2.1 Masalah: Multi-User, Multi-Business	5
2.2 Solusi: 2 Strategi Authentication	5
2.3 Strategi 1: Demo Mode (Recommended untuk Hackathon)	6
2.3.1 Cara Kerja Demo Mode	6

2.4	Strategi 2: Full Authentication	7
2.4.1	Cara Kerja Full Authentication	7
2.5	Decision Tree: Pilih Strategi Mana?	8
2.6	Recommendation untuk Tim	9
2.7	Argumen untuk Juri	9
3	Database Schema dan Dataset	11
3.1	Overview Database	11
3.2	Schema 1: Demo Mode (Tanpa Authentication)	11
3.2.1	Tabel: products	11
3.2.2	Tabel: sales	12
3.3	Schema 2: Full Authentication (Multi-User)	12
3.3.1	Tabel Tambahan: users	12
3.3.2	Update Tabel: products (dengan user_id)	12
3.3.3	Update Tabel: sales (dengan user_id)	13
3.4	Query Patterns: Demo Mode vs Full Auth	13
3.4.1	Demo Mode: Get All Products	13
3.4.2	Full Auth: Get User's Products Only	13
3.5	Dataset Requirements untuk Demo	13
3.6	Sample Data Generation Strategy	14
3.6.1	5 Produk untuk Demo	14
3.6.2	Pola Data yang Realistik	14
3.6.3	Contoh Data untuk 1 Produk, 5 Hari	15
3.6.4	Cara Generate Dataset	15
3.7	Dataset Checklist untuk Hackathon	16
4	Analisis Rubrik Penilaian	17
4.1	Komposisi Lengkap	17
4.2	Insight Strategis dari Bobot	17
4.3	Kategori 1: Code Quality (10 Poin - 5%)	18
4.3.1	Breakdown Sub-kriteria	18
4.3.2	Strategi Dapat 10/10	18
4.4	Kategori 2: Architecture (20 Poin - 10%)	19
4.4.1	Breakdown Sub-kriteria	19
4.4.2	Strategi Dapat 20/20	19
4.5	Kategori 3: Innovation (40 Poin - 20%)	20
4.5.1	Breakdown Sub-kriteria	20
4.5.2	Argumen Kebaruan Ide (20 poin)	20
4.5.3	Argumen Kompleksitas Teknis (20 poin)	21
4.6	Kategori 4: Functionality (50 Poin - 25%)	22
4.6.1	Breakdown Sub-kriteria	22
4.6.2	6 Fitur Utama yang HARUS Berfungsi	22
4.6.3	Happy Path untuk Demo	23
4.6.4	Stabilitas & UX Checklist	24
4.7	Kategori 5: Documentation & Video Demo (80 Poin - 40%)	24
4.7.1	Breakdown Sub-kriteria	25
4.7.2	Video Demo: Struktur (4-5 menit)	25
4.7.3	Script Narasi Hook (Contoh)	26
4.7.4	README.md Template (Struktur)	26
4.8	Bonus Points (+20 Poin)	27
4.8.1	Bonus 1: Advanced Tech (+10 poin)	27
4.8.2	Bonus 2: Deployment (+10 poin)	27

4.9 Penalti yang WAJIB Dihindari	27
4.9.1 Checklist Anti-Penalti	28
II FORMULA MATEMATIS	29
5 Overview Sistem Analytics	29
5.1 Notasi yang Digunakan	29
6 Formula 1: Calendar Factors	29
6.1 Day-of-Week Factor	29
6.2 Payday Window Factor	30
6.3 Special Day Factor	30
7 Formula 2: Expected Demand	30
8 Formula 3: Exponential Moving Average (EMA)	31
9 Formula 4: Multi-Window Momentum	31
10 Formula 5: Adaptive Burst Score	32
11 Formula 6: Burst Classification	33
12 Formula 7: Priority Score	34
III PEMBAGIAN TUGAS TIM	35
13 Role Assignment	35
14 Timeline Detail: Hari per Hari	35
14.1 HARI 1: Foundation & Setup	35
14.2 HARI 2: Core Backend	36
14.3 HARI 3: Core Analytics	37
14.4 HARI 4: Features Complete	38
14.5 HARI 5: Integration & Polish	39
14.6 HARI 6: Documentation & Video Prep	40
14.7 HARI 7: Video & Submit	41
IV FINAL CHECKLIST	42
15 Pre-Submit Checklist	42
15.1 Repository	42
15.2 Deployment	42
15.3 Video	42
15.4 Functionality	43
16 Target Score Final	43

Part I

KONSEP DAN STRATEGI

1 Konsep Aplikasi: Dashboard Analytics, BUKAN Chatbot

WAJIB - KRITIS

PENTING: Ini BUKAN Chatbot!

Banyak yang salah paham karena ada history idea "WarungGPT via WhatsApp" di awal brainstorming. Tapi konsep FINAL adalah:

Web Dashboard Analytics (Progressive Web App)

BUKAN conversational AI, BUKAN chatbot, BUKAN voice assistant!

1.1 Apa itu AI Market Pulse Radar?

KEY INSIGHT

AI Market Pulse Radar adalah aplikasi web dashboard yang membantu UMKM Indonesia memahami tren penjualan mereka melalui analisis otomatis berbasis AI.

Konsep Utama:

- User input data penjualan via FORM (bukan chat!)
- AI menganalisis secara OTOMATIS di background
- Hasil ditampilkan dalam bentuk VISUAL (chart, card, badge)
- User membaca insight dan mengambil keputusan

Analogi Produk:

- Mirip: Google Analytics, Tableau, PowerBI
- BUKAN mirip: ChatGPT, WhatsApp Bot, Siri

1.2 Kenapa BUKAN Chatbot?

PENTING

History Singkat:

Chat Awal (28 Nov):

- Ada ide "WarungGPT" - asisten via WhatsApp
- Konsep: input via suara, chat, foto
- WhatsApp-first approach

Pivot (30 Nov):

- Ternyata WhatsApp TIDAK DIPERBOLEHKAN di rules hackathon
- Switch ke Progressive Web App (PWA)
- Dari conversational → Dashboard visual

Konsep Final:

- Web dashboard dengan analytics
- Rule-based AI (bukan LLM)
- Visual interface (bukan conversational)

1.3 Perbandingan: Chatbot vs Dashboard

Aspek	Chatbot (BUKAN ini!)	Dashboard (INI yang kita buat!)
Interface	Chat bubble, text input	Form, button, chart, table
Input Method	Ketik pesan, voice	Isi form, klik button
AI Type	NLP, LLM (GPT-like)	Rule-based analytics
Output	Text response	Chart, badge, alert card
Interaction	“Produk apa yang trending?”	Buka halaman Ranking → Lihat table
Example	ChatGPT, WhatsApp Bot	Google Analytics, Tableau

1.4 User Flow: Bagaimana User Berinteraksi?**CONTOH**

**Skenario: User ingin tahu produk mana yang sedang trending
SALAH (Chatbot):**

1. User ketik: "Produk apa yang lagi trending?"
2. AI balas: "Berdasarkan analisis, Nasi Goreng sedang trending..."
3. User ketik lagi: "Kenapa bisa trending?"
4. AI balas: "Karena penjualan naik 45%..."

BENAR (Dashboard):

1. User buka halaman "Dashboard" atau "Ranking"
2. Langsung LIHAT tabel ranking produk
3. Produk 1 (Nasi Goreng) ada badge TRENDING
4. KLIK produk → Lihat detail: chart trend, momentum score
5. Baca recommendations di card

1.5 Flow Lengkap dari Awal sampai Akhir**LANGKAH 1: Landing Page**

- User buka aplikasi (URL production atau localhost)
- Lihat landing page dengan penjelasan singkat produk
- Klik tombol "Mulai" atau "Dashboard"

LANGKAH 2: Setup Produk (first time)

- Masuk ke halaman Products
- Klik tombol "Tambah Produk"
- Isi form: Nama produk, kategori, harga (optional)
- Klik "Simpan"
- Produk muncul di list
- Ulangi untuk 4-5 produk

LANGKAH 3: Input Penjualan Harian

- Setiap hari, user masuk ke halaman "Input Penjualan"
- Lihat form dengan list semua produk
- Input jumlah terjual untuk setiap produk
- Klik "Simpan"
- Sistem menampilkan toast "Data berhasil disimpan"

LANGKAH 4: Lihat Dashboard Summary

- Buka halaman "Dashboard"
- Lihat summary cards: Total penjualan hari ini, Revenue, Comparison vs kemarin
- Jika ada BURST, muncul Alert Card merah: " Nasi Goreng naik 85% dari biasanya!"
- Lihat quick ranking: Top 3 produk

LANGKAH 5: Lihat Product Detail

- Klik alert atau klik produk dari dashboard
- Masuk ke halaman detail produk
- Lihat chart trend 30 hari (line chart)
- Lihat badge status: TRENDING UP, STABLE, DECLINING
- Lihat metrics: Momentum score, Burst score
- Baca recommendations di card insight

LANGKAH 6: Lihat Product Ranking

- Masuk ke halaman "Products" atau "Ranking"
- Lihat tabel semua produk ter-ranking berdasarkan priority score
- Produk dengan priority tinggi di atas
- Setiap produk ada badge status dan momentum indicator

LANGKAH 7: Lihat Weekly Report (Optional)

- Masuk ke halaman "Reports"
- Lihat ringkasan minggu lalu
- Top performers, insights, recommendations

2 Authentication Strategy

2.1 Masalah: Multi-User, Multi-Business

WAJIB - KRITIS

Problem Statement:

Setiap user punya bisnis berbeda dengan data berbeda:

- User A = Warung Nasi Goreng Pak Budi (jual: Nasi Goreng, Mie Ayam, Es Teh)
- User B = Toko Kopi Bu Susi (jual: Espresso, Latte, Cappuccino)
- User C = Toko Sepatu Mas Joko (jual: Sneakers, Sandal, Boots)

Data mereka HARUS TERPISAH!

Pertanyaan:

- Apakah user harus login?
- Bagaimana sistem tahu data siapa yang ditampilkan?
- Bagaimana mencegah User A lihat data User B?

2.2 Solusi: 2 Strategi Authentication

DECISION POINT

Ada 2 pilihan strategi untuk hackathon 7 hari:

Strategi 1: Demo Mode (NO AUTHENTICATION)

- Semua orang akses URL yang sama
- Lihat data DEMO yang sama (contoh: Warung Pak Budi)
- TIDAK ada login/register
- Fokus showcase AI analytics

Strategi 2: Full Authentication

- User harus register/login
- Setiap user punya data terpisah
- User A tidak bisa lihat data User B
- Production-ready

Pilihan mana? → Lihat section Decision Tree di bawah!

2.3 Strategi 1: Demo Mode (Recommended untuk Hackathon)

WINNING FACTOR

Kenapa Demo Mode bagus untuk hackathon?

Advantages:

1. **Hemat waktu development** - Tidak perlu 3-4 jam untuk setup auth system
2. **Fokus ke core feature** - AI analytics adalah yang dinilai, bukan auth
3. **Demo lebih smooth** - Juri bisa langsung test tanpa ribet sign up
4. **Deployment gampang** - No user management, no session handling
5. **Video demo lebih fokus** - Tidak perlu explain login flow

Disadvantages:

1. Tidak production-ready (semua user lihat data yang sama)
2. Tidak bisa showcase multi-tenancy
3. Kurang realistik untuk real-world usage

Kapan pakai strategi ini?

- Timeline ketat (7 hari)
- Tim fokus ke algoritma dan analytics
- Core features belum 100% selesai di Hari 4-5

2.3.1 Cara Kerja Demo Mode

User Flow:

1. User buka URL: <https://market-pulse.vercel.app>
2. LANGSUNG masuk ke dashboard (no login screen)
3. Lihat data demo "Warung Nasi Goreng Pak Budi"
4. Input penjualan → Data tersimpan di database shared
5. Semua user lihat data yang sama

Database:

- Tidak ada tabel users
- Tidak ada kolom user.id di products/sales
- Semua data shared

Di Video Demo:

"Ini adalah demo untuk Warung Nasi Goreng Pak Budi. Dalam versi production, setiap user akan memiliki akun terpisah dengan data bisnis masing-masing. Untuk hackathon ini, kami fokus showcase AI analytics engine yang menjadi core innovation kami."

2.4 Strategi 2: Full Authentication

PENTING

Kapan pakai strategi ini?

Gunakan Full Auth jika:

1. Core features SUDAH 100% jalan di Hari 4-5
2. Tim masih punya energi dan waktu (Hari 6)
3. Ingin showcase production-ready application
4. Ingin bonus poin dari juri (lebih polished)

Technology Stack:

- NextAuth.js - Authentication library untuk Next.js
- JWT Session - Token-based session management
- Credentials Provider - Email/password login

Effort Estimate:

- Setup NextAuth: 30 menit
- Update database schema: 30 menit
- Update API routes dengan auth check: 1-2 jam
- Update frontend dengan login page: 1 jam
- Testing: 30 menit
- **Total: 3-4 jam**

2.4.1 Cara Kerja Full Authentication

User Flow:

1. User buka URL: <https://market-pulse.vercel.app>
2. Redirect ke login page
3. **Option A:** Login dengan email/password (jika sudah punya akun)
4. **Option B:** Register akun baru (nama bisnis, email, password)
5. Setelah login → Redirect ke dashboard
6. Lihat data MILIK USER SENDIRI
7. User lain login → Lihat data mereka sendiri (terpisah)

Demo Users (Hardcoded untuk Demo):

Untuk kemudahan demo di video, bisa hardcode 2-3 user:

- Email: budi@warung.com — Password: demo123 — Bisnis: Warung Nasi Goreng
- Email: susi@kopi.com — Password: demo123 — Bisnis: Kopi Kenangan Susi

- Email: joko@sepatu.com — Password: demo123 — Bisnis: Toko Sepatu Joko

Session Management:

- Login → Dapat JWT token
- Token tersimpan di cookie
- Setiap API request → Check token
- Token invalid → Redirect ke login
- Logout → Hapus token

2.5 Decision Tree: Pilih Strategi Mana?

DECISION POINT**DECISION TREE**

Pertanyaan 1: Apakah SEMUA core features (analytics, chart, ranking, recommendations) sudah 100% jalan dan tested di Hari 4-5?

TIDAK → Pakai **Demo Mode**

- Fokus finalize core features dulu
- Authentication bukan prioritas untuk penilaian
- Hemat waktu untuk polish dan video

YA → Lanjut ke Pertanyaan 2

Pertanyaan 2: Apakah tim masih punya energi dan waktu di Hari 6?

TIDAK → Pakai **Demo Mode**

- Better safe than sorry
- Fokus ke documentation dan video
- Polish UI/UX lebih penting

YA → Lanjut ke Pertanyaan 3

Pertanyaan 3: Apakah ada anggota tim yang sudah familiar dengan NextAuth.js atau authentication system?

TIDAK → **HATI-HATI!**

- Learning curve bisa makan waktu
- Risk tinggi untuk last-minute implementation
- Recommended: Tetap pakai Demo Mode

YA → **GO FOR IT!**

- Implement Full Authentication
- Allocate 3-4 jam di Hari 6
- Bonus: Aplikasi lebih production-ready

2.6 Recommendation untuk Tim

WINNING FACTOR

Rekomendasi Kami:

Hari 1-5: START dengan Demo Mode

- Build semua core features tanpa auth
- Test dengan single shared database
- Deploy ke Vercel
- Prepare video demo dengan akun demo

Hari 6 (Morning): Evaluate

- Semua features jalan? Bug critical sudah fix?
- Tim masih fresh atau sudah capek?
- Ada yang bisa handle auth implementation?

Hari 6 (Afternoon): Decision

- **Jika YA semua** → Implement auth (3-4 jam)
- **Jika ADA yang NO** → Skip auth, polish existing features

Mantra:

“Better to have PERFECT demo mode than BUGGY full auth!”

2.7 Argumen untuk Juri

Jika ditanya: ”Kenapa tidak ada authentication?”

CONTOH

Jawaban yang Baik:

”Untuk hackathon ini, kami memutuskan untuk fokus mengembangkan core innovation yaitu AI analytics engine yang context-aware dan adaptive. Kami menggunakan demo mode untuk memudahkan evaluasi fitur-fitur analytics kami.

Untuk production deployment, kami sudah merencanakan implementasi session-based authentication dengan user separation, dimana setiap user akan memiliki dashboard dan data terpisah sesuai bisnis mereka masing-masing.

Architecture kami sudah di-design dengan separation of concerns yang baik, sehingga menambahkan authentication layer hanya perlu 3-4 jam implementation tanpa mengubah core logic.”

Jika sudah implement authentication:

CONTOH

Jawaban yang Baik:

”Kami sudah implement session-based authentication menggunakan NextAuth.js dengan JWT tokens. Setiap user memiliki akun terpisah dan hanya bisa mengakses data bisnis mereka sendiri.

Untuk kemudahan demo, kami sudah menyiapkan beberapa demo accounts dengan berbagai jenis bisnis:

- Warung Nasi Goreng - untuk showcase F&B use case
- Toko Kopi - untuk showcase beverage business
- Toko Sepatu - untuk showcase retail use case

Database kami fully support multi-tenancy dengan proper data isolation.”

3 Database Schema dan Dataset

3.1 Overview Database

KEY INSIGHT

Database Minimum untuk Hackathon:

Hanya butuh **2-3 tabel** saja untuk semua fitur jalan!

Mandatory (WAJIB):

1. **products** - Daftar produk yang dijual
2. **sales** - Data penjualan harian

Optional (kalau pakai Full Auth):

1. **users** - Data user/pemilik bisnis

Technology Options:

- Vercel Postgres (Recommended - free tier, serverless)
- SQLite (Simplest - file-based, no setup)
- Supabase (Alternative - free tier, has GUI)

3.2 Schema 1: Demo Mode (Tanpa Authentication)

3.2.1 Tabel: products

Column	Type	Description
id	INTEGER	Primary key, auto increment
name	VARCHAR(100)	Nama produk, contoh: "Nasi Goreng", "Mie Ayam"
category	VARCHAR(50)	Kategori produk, contoh: "Makanan", "Minuman" (Optional)
price	DECIMAL(10,2)	Harga jual per unit, contoh: 15000 (Optional)
created_at	TIMESTAMP	Kapan produk ditambahkan

Contoh Data:

id	name	category	price
1	Nasi Goreng	Makanan	15000
2	Mie Ayam	Makanan	12000
3	Es Teh	Minuman	3000
4	Es Jeruk	Minuman	5000
5	Soto Ayam	Makanan	13000

3.2.2 Tabel: sales

Column	Type	Description
id	INTEGER	Primary key, auto increment
product_id	INTEGER	Foreign key ke products.id
date	DATE	Tanggal penjualan, format: YYYY-MM-DD
quantity	INTEGER	Jumlah unit terjual
revenue	DECIMAL(10,2)	Total revenue (quantity × price), Optional
has_promo	BOOLEAN	Apakah ada promo di hari tersebut (Optional)
created_at	TIMESTAMP	Kapan data diinput

Contoh Data:

id	product_id	date	quantity	revenue	has_promo
1	1	2024-11-01	45	675000	false
2	1	2024-11-02	50	750000	false
3	1	2024-11-03	48	720000	false
4	1	2024-11-04	95	1425000	false
5	2	2024-11-01	30	360000	false
...

3.3 Schema 2: Full Authentication (Multi-User)

3.3.1 Tabel Tambahan: users

Column	Type	Description
id	INTEGER	Primary key, auto increment
email	VARCHAR(100)	Email user, UNIQUE
password_hash	VARCHAR(255)	Hashed password (NEVER store plain password!)
name	VARCHAR(100)	Nama pemilik bisnis
business_name	VARCHAR(100)	Nama bisnis, contoh: "Warung Nasi Pak Budi"
created_at	TIMESTAMP	Kapan akun dibuat

Contoh Data:

id	email	name	business_name
1	budi@warung.com	Pak Budi	Warung Nasi Goreng
2	susi@kopi.com	Bu Susi	Kopi Kenangan Susi
3	joko@sepatu.com	Mas Joko	Toko Sepatu Joko

3.3.2 Update Tabel: products (dengan user_id)

Column	Type	Description
id	INTEGER	Primary key, auto increment
user_id	INTEGER	Foreign key ke users.id (TAMBAHAN!)
name	VARCHAR(100)	Nama produk
category	VARCHAR(50)	Kategori produk (Optional)
price	DECIMAL(10,2)	Harga jual (Optional)
created_at	TIMESTAMP	Kapan produk ditambahkan

Contoh Data dengan Multi-User:

id	user_id	name	category	price
1	1	Nasi Goreng	Makanan	15000
2	1	Mie Ayam	Makanan	12000
3	2	Espresso	Minuman	25000
4	2	Latte	Minuman	30000
5	3	Sneakers Nike	Sepatu	500000

3.3.3 Update Tabel: sales (dengan user_id)

Column	Type	Description
id	INTEGER	Primary key, auto increment
user_id	INTEGER	Foreign key ke users.id (TAMBAHAN!)
product_id	INTEGER	Foreign key ke products.id
date	DATE	Tanggal penjualan
quantity	INTEGER	Jumlah unit terjual
revenue	DECIMAL(10,2)	Total revenue (Optional)
has_promo	BOOLEAN	Ada promo atau tidak (Optional)
created_at	TIMESTAMP	Kapan data diinput

3.4 Query Patterns: Demo Mode vs Full Auth

3.4.1 Demo Mode: Get All Products

Konsep: Ambil semua produk dari database (shared untuk semua user)

Query Pattern:

- SELECT * FROM products
- Tidak ada filter user_id
- Semua user lihat produk yang sama

3.4.2 Full Auth: Get User's Products Only

Konsep: Ambil hanya produk milik user yang sedang login

Query Pattern:

- Get user_id dari session (contoh: user_id = 1 untuk Pak Budi)
- SELECT * FROM products WHERE user_id = 1
- User 1 hanya lihat produk mereka sendiri
- User 2 lihat produk user 2, dst.

Data Isolation:

- Pak Budi (user_id=1) hanya lihat: Nasi Goreng, Mie Ayam
- Bu Susi (user_id=2) hanya lihat: Espresso, Latte
- Mas Joko (user_id=3) hanya lihat: Sneakers Nike
- Tidak ada cross-contamination!

3.5 Dataset Requirements untuk Demo

PENTING

Berapa banyak data yang dibutuhkan?

Minimum untuk MVP:

- **Products:** 4-5 produk
- **Sales:** 30 hari terakhir \times 4-5 produk = 120-150 rows

Ideal untuk Demo Showcase:

- **Products:** 5 produk
- **Sales:** 60 hari \times 5 produk = 300 rows

Kenapa 60 hari?

- Formula momentum butuh 30 hari untuk baseline
- Masih punya 30 hari extra untuk validation
- Cukup untuk showcase seasonal patterns
- Chart trend 30 hari terlihat smooth

3.6 Sample Data Generation Strategy

3.6.1 5 Produk untuk Demo

ID	Name	Category	Price	Baseline/day
1	Nasi Goreng	Makanan	15000	50 porsi
2	Mie Ayam	Makanan	12000	40 porsi
3	Es Teh	Minuman	3000	120 gelas
4	Es Jeruk	Minuman	5000	80 gelas
5	Soto Ayam	Makanan	13000	35 porsi

Baseline = Rata-rata penjualan normal per hari

3.6.2 Pola Data yang Realistik

1. Calendar Factors (Otomatis dari Formula)

- Weekend (Sabtu-Minggu) \rightarrow Naik 30%
- Payday window (tgl 25-5) \rightarrow Naik 30%
- Weekday normal \rightarrow Baseline

2. Random Variation

- Setiap hari ada variasi natural $\pm 10\%$
- Biar realistic, not too perfect

3. Burst Events (untuk Trigger Alert)

- 1-2 produk punya burst dalam 60 hari
- Burst = Naik 80-100% dari expected

- Contoh: Nasi Goreng naik karena viral di TikTok

4. Declining Trend (Optional)

- 1 produk show declining trend
- Contoh: Soto Ayam turun gradual karena kompetitor baru

3.6.3 Contoh Data untuk 1 Produk, 5 Hari

Nasi Goreng (Baseline = 50 porsi/hari):

Date	Day	Quantity	Reasoning
2024-11-25	Senin	48	Normal weekday (-4% variation)
2024-11-26	Selasa	52	Normal weekday (+4% variation)
2024-11-27	Rabu	85	Payday + Weekend = 50 × 1.3 × 1.3 = 85
2024-11-28	Kamis	95	BURST! Viral di TikTok (+90%)
2024-11-29	Jumat	88	Masih tinggi (post-viral effect)

3.6.4 Cara Generate Dataset

Option 1: Manual di Spreadsheet

1. Buat Google Sheets / Excel
2. Kolom: product_id, date, quantity
3. Isi manual dengan pattern yang realistic
4. Export ke CSV
5. Import ke database

Option 2: Script Otomatis (Python/JavaScript)

1. Buat script dengan logic calendar factors
2. Loop 60 hari, 5 produk
3. Generate quantity dengan variation + burst events
4. Output ke CSV atau langsung ke database
5. Lebih cepat dan consistent

Option 3: Seed Script di Backend

1. Buat seed script di Next.js API route
2. Call sekali untuk populate database
3. Good for development and testing

3.7 Dataset Checklist untuk Hackathon

Item	Quantity	Notes
Products	5 items	Mix makanan dan minuman
Sales Records	300 rows	60 hari × 5 produk
Date Range	60 days	2024-10-01 s/d 2024-11-30
Burst Events	5-10 times	Spread across products/dates
Declining Product	1 item	Show algorithm can detect downtrend
Weekend Data	16 days	Sabtu-Minggu dalam 60 hari
Payday Data	24 days	Tanggal 25-31 + 1-5

WAJIB - KRITIS

WAJIB untuk Demo Video:

Pastikan ada minimal 1 produk yang:

1. Sedang BURST di tanggal demo (untuk trigger alert merah)
2. Ada TRENDING badge (momentum positif)
3. Bisa showcase semua classification: VIRAL, SEASONAL, STABLE

Ini penting untuk demo video yang WOW!

4 Analisis Rubrik Penilaian

WAJIB - KRITIS

TOTAL SKOR MAKSIMAL: 220 PONI

- Skor Dasar: 200 poin
- Bonus Maksimal: +20 poin
- Penalti bisa mengurangi skor (bahkan DISKUALIFIKASI)

Rubrik ini fokus pada **Product & Pitch** — artinya kemampuan MENJUAL ide sama pentingnya dengan kualitas teknis!

4.1 Komposisi Lengkap

No	Kategori	Bobot	Poin	Prioritas
1	Code Quality	5%	10	Low
2	Architecture	10%	20	Medium
3	Innovation	20%	40	High
4	Functionality	25%	50	High
5	Documentation & Video Demo	40%	80	HIGHEST
Subtotal Dasar		200		
6	Bonus: Advanced Tech	-	+10	Must Get
7	Bonus: Deployment	-	+10	Must Get
TOTAL MAKSIMAL		220		

4.2 Insight Strategis dari Bobot

KEY INSIGHT

Observasi Penting:

1. **40% di Documentation & Video** = Ini yang PALING menentukan! Tim yang video-nya bagus tapi kode biasa-biasa bisa MENANG dari tim yang kodennya bagus tapi video jelek.
2. **Code Quality cuma 5%** = Jangan habiskan waktu untuk perfect code. Yang penting BERSIH dan JALAN.
3. **Bonus +20 itu BESAR** = Setara dengan seluruh poin Architecture. WAJIB dapat bonus ini!
4. **Penalti bisa fatal** = Satu kesalahan (API key di repo) bisa kurangi 20 poin. Hati-hati!

4.3 Kategori 1: Code Quality (10 Poin - 5%)

4.3.1 Breakdown Sub-kriteria

Sub-kriteria	Checklist	Poin
Kebersihan Dasar	<ul style="list-style-type: none"> • Penamaan variabel JELAS (bukan x, a, tmp) • TIDAK ada dead code • TIDAK ada console.log sampah • Indentasi RAPI dan konsisten 	0-5
Best Practice Dasar	<ul style="list-style-type: none"> • TIDAK ada hardcoded credentials (API Key) • Struktur file TIDAK berantakan • TIDAK semua file di root folder 	0-5

4.3.2 Strategi Dapat 10/10

1. Penamaan Variabel yang Jelas

Contoh SALAH:

- const x = getSales()
- const a = x.filter(i => i.qty > 0)
- let tmp = 0

Contoh BENAR:

- const salesData = getSalesData()
- const validSales = salesData.filter(sale => sale.quantity > 0)
- let totalQuantity = 0

2. Hapus Dead Code dan Console.log

Sebelum submit, WAJIB:

- Search semua console.log dan hapus (kecuali error handling)
- Hapus commented code blocks
- Hapus unused imports
- Hapus TODO comments

3. File .gitignore yang WAJIB

Must include:

- node_modules/
- .env (SANGAT PENTING!)

- .next/
- build/
- npm-debug.log*

4. File .env.example

WAJIB buat template tanpa nilai asli:

- DATABASE_URL=postgresql://user:password@host:5432/dbname
- NEXTAUTH_SECRET=your-secret-here
- File ini DI-COMMIT, tapi .env yang asli TIDAK!

4.4 Kategori 2: Architecture (20 Poin - 10%)

4.4.1 Breakdown Sub-kriteria

Sub-kriteria	Checklist	Poin
Desain Sistem	<ul style="list-style-type: none"> • Pemisahan LOGIS (UI terpisah dari logic bisnis) • Aliran data JELAS dan tidak berbelit-belit 	0-10
Tech Stack	<ul style="list-style-type: none"> • Teknologi TEPAT GUNA (tidak overkill, tidak obsolete) • Library eksternal digunakan secara EFEKTIF 	0-10

4.4.2 Strategi Dapat 20/20

Kenapa Next.js adalah pilihan TEPAT GUNA:

- Fullstack dalam satu framework (frontend + API)
- Deploy ke Vercel dalam hitungan menit (bonus +10!)
- TIDAK overkill (bukan microservices untuk hackathon)
- TIDAK obsolete (versi terbaru, actively maintained)
- Ecosystem besar, dokumentasi lengkap

Prinsip Separation of Concerns:

1. **app/** = HANYA routing dan page structure
2. **components/** = HANYA UI rendering (no business logic!)
3. **lib/analytics/** = HANYA algoritma dan perhitungan
4. **lib/database/** = HANYA operasi database
5. **app/api/** = HANYA handle HTTP request/response

Contoh SALAH: Menaruh perhitungan momentum di dalam component React.

Contoh BENAR: Component hanya memanggil API, API memanggil lib/analytics.

Tech Stack Final:

Layer	Technology	Justifikasi (untuk juri)
Framework	Next.js 14	Fullstack, modern, easy deploy
Language	TypeScript	Type safety, better DX
Styling	Tailwind CSS	Rapid development, no CSS files
Charts	Recharts	React-native, declarative
Database	Vercel Postgres	Free tier, serverless
Deployment	Vercel	Free, instant, auto-HTTPS

4.5 Kategori 3: Innovation (40 Poin - 20%)

4.5.1 Breakdown Sub-kriteria

Sub-kriteria	Checklist	Poin
Kebaruan Ide	<ul style="list-style-type: none"> Solusi UNIK, bukan clone aplikasi existing Pendekatan KREATIF untuk masalah tema hackathon 	0-20
Kompleksitas Teknis	<ul style="list-style-type: none"> Implementasi fitur SULIT (AI, algoritma kompleks) BUKAN sekadar CRUD sederhana 	0-20

4.5.2 Argumen Kebaruan Ide (20 poin)

WINNING FACTOR

Apa yang membuat ide kita UNIK:

1. Context-Aware Analytics

- Bukan generic dashboard yang hanya tampilkan angka
- PAHAM konteks Indonesia: payday cycle, Ramadan, weekend pattern
- Sistem "tahu" bahwa Sabtu tanggal 27 harusnya ramai

2. Proactive Intelligence

- Bukan reactive (user tanya, sistem jawab)
- PROAKTIF memberitahu saat ada anomaly
- Memberikan REKOMENDASI, bukan hanya data

3. UMKM Mikro Focus

- Bukan untuk enterprise dengan tim data science
- Untuk warung, PKL, toko kecil yang punya data tapi tidak bisa membaca

- Simple interface, bahasa Indonesia, actionable insight

Bukan Clone Dari:

- Bukan Tableau/PowerBI (terlalu kompleks untuk UMKM)
- Bukan POS system (kita fokus ke analytics, bukan transaksi)
- Bukan generic dashboard template

4.5.3 Argumen Kompleksitas Teknis (20 poin)**WINNING FACTOR****Fitur-fitur yang BUKAN sekadar CRUD:****1. Multi-Window Momentum Analysis**

- Exponential Moving Average (EMA) dengan 3 window
- Weighted combination untuk smooth trend detection
- Ini teknik yang dipakai di financial analysis

2. Adaptive Burst Detection

- Bukan Z-score biasa terhadap mean
- Context-aware: bandingkan dengan EXPECTED demand
- Menghindari false positive saat seasonal peak

3. Rule-Based Classification Engine

- Decision tree untuk klasifikasi burst type
- Mempertimbangkan duration, calendar, promo flag
- Output: VIRAL, PROMO, SEASONAL, NOISE

4. Calendar-Adjusted Expected Demand

- Day-of-week factors learned dari data
- Payday window detection (tanggal 25-5)
- Special day handling (Ramadan, libur)

5. Priority Scoring System

- Kombinasi momentum + burst dengan weighting
- Normalisasi untuk fair comparison
- Automatic ranking

Ini adalah EXPERT SYSTEM, bukan CRUD!

4.6 Kategori 4: Functionality (50 Poin - 25%)

4.6.1 Breakdown Sub-kriteria

Sub-kriteria	Checklist	Poin
Fitur Utama	<ul style="list-style-type: none"> Fitur yang DIJANJIKAN di deskripsi BER-JALAN dengan baik Alur utama (Happy Path) TUNTAS dari awal sampai akhir 	0-30
Stabilitas & UX	<ul style="list-style-type: none"> MINIM bug/error saat demo RESPONSIF dan interaksi nyaman digunakan 	0-20

4.6.2 6 Fitur Utama yang HARUS Berfungsi

WAJIB - KRITIS

WAJIB SEMUA JALAN saat demo:

1. Setup Produk

- User bisa tambah produk yang dijual
- User bisa edit/hapus produk

2. Input Penjualan Harian

- User bisa input jumlah penjualan per produk
- Support input untuk tanggal tertentu
- Validasi input (tidak boleh negatif)

3. Dashboard Summary

- Tampilkan ringkasan hari ini
- Total penjualan, revenue (jika ada harga)
- Perbandingan vs kemarin

4. Burst Alert

- Deteksi anomaly secara otomatis
- Tampilkan alert card jika ada burst
- Klasifikasi burst type

5. Product Detail & Trend

- Chart trend 30 hari per produk
- Tampilkan momentum score
- Tampilkan recommendations

6. Product Ranking

- Ranking berdasarkan priority score
- Tampilkan status (trending, stable, declining)

4.6.3 Happy Path untuk Demo

Skenario demo yang HARUS lancar tanpa error:

1. Landing Page

- Buka aplikasi
- Lihat landing page dengan penjelasan singkat
- Klik "Mulai" atau "Dashboard"

2. Setup Produk (jika pertama kali)

- Tambah 4-5 produk: Nasi Goreng, Mie Ayam, Es Teh, Es Jeruk
- Produk tersimpan dan muncul di list

3. Input Penjualan

- Masuk ke halaman input
- Input penjualan untuk hari ini:
 - Nasi Goreng: 95 (tinggi - untuk trigger burst)
 - Mie Ayam: 45
 - Es Teh: 120
 - Es Jeruk: 80
- Klik simpan, data tersimpan

4. Lihat Dashboard

- Dashboard menampilkan summary
- ALERT muncul untuk Nasi Goreng (burst detected!)
- Tampilkan total, revenue, comparison

5. Klik Alert / Product Detail

- Lihat detail Nasi Goreng
- Chart trend 30 hari
- Momentum: +45% (TRENDING)
- Burst Score: 3.2
- Recommendations muncul

6. Lihat Ranking

- Semua produk ter-ranking
- Nasi Goreng di posisi #1 (priority tinggi)
- Badge status: TRENDING, STABLE, dll

7. Weekly Report (Optional)

- Ringkasan minggu lalu
- Top performers
- Insights

4.6.4 Stabilitas & UX Checklist

Aspek	Checklist
Loading States	<ul style="list-style-type: none"> • Skeleton loading saat fetch data • Button disabled saat submit • Loading indicator yang jelas
Error States	<ul style="list-style-type: none"> • Error message yang user-friendly • Tidak crash, tampilkan fallback • Retry option jika gagal
Empty States	<ul style="list-style-type: none"> • Message jika belum ada data • CTA untuk action berikutnya • Tidak blank page
Responsive	<ul style="list-style-type: none"> • Berfungsi di mobile (375px+) • Berfungsi di tablet • Berfungsi di desktop
Feedback	<ul style="list-style-type: none"> • Success toast setelah save • Hover states pada button • Active states pada navigation

4.7 Kategori 5: Documentation & Video Demo (80 Poin - 40%)

WAJIB - KRITIS

INI KATEGORI TERBESAR!

40% dari total skor ada di sini. Alokasikan waktu yang SIGNIFIKAN (minimal 1.5 hari) untuk dokumentasi dan video!

4.7.1 Breakdown Sub-kriteria

Sub-kriteria	Checklist	Poin
Video Demo: Story-telling	<ul style="list-style-type: none"> • Problem & Solution dijelaskan SANGAT JELAS • Alur cerita MENARIK • Audio JELAS • Visual MENDUKUNG 	0-30
Kualitas Demo Produk	<ul style="list-style-type: none"> • Video menampilkan aplikasi BERJALAN REAL • Bukan mockup/Figma • Fitur unggulan didemokan JELAS dalam durasi 	0-25
Dokumentasi Teknis (README)	<ul style="list-style-type: none"> • WAJIB: Instruksi instalasi langkah-demi-langkah • Penjelasan fitur dan cara penggunaan • Screenshots aplikasi di dalam README 	0-25

4.7.2 Video Demo: Struktur (4-5 menit)

Timeline Recommended:

Time	Section	Content
00:00-00:05	Opening	Logo/Title card
00:05-00:35	Hook + Problem	Cerita Pak Budi, Visual warung
00:35-01:05	Problem Deepdive	3 masalah UMKM Indonesia
01:05-01:30	Solution Intro	Perkenalkan produk, tagline
01:30-03:30	Demo Aplikasi REAL	Tunjukkan semua fitur utama
03:30-04:00	Tech Highlight	Jelaskan AI yang digunakan
04:00-04:20	Impact	Potensi dampak untuk UMKM
04:20-04:40	Closing	Tagline ulang, team credits
04:40-05:00	End Card	Logo, URL, terima kasih

4.7.3 Script Narasi Hook (Contoh)

CONTOH

[00:05 - 00:35] HOOK + PROBLEM

"Pak Budi punya warung Nasi Goreng di Bekasi. Setiap hari dia mencatat penjualan di buku tulis.

Minggu lalu, ada food vlogger yang mampir dan upload video. Dalam 3 hari, video itu dapat 500 ribu views.

Tapi Pak Budi baru sadar warungnya viral... setelah kehabisan beras di hari kedua.

Ratusan customer datang dan kecewa. Pak Budi kehilangan jutaan rupiah potential revenue.

Masalahnya bukan Pak Budi tidak punya data. Dia PUNYA catatan penjualan setiap hari.

Tapi catatan itu tidak pernah berbicara ke dia."

4.7.4 README.md Template (Struktur)

Sections yang WAJIB ada:

1. Title & Description

- Nama project
- Tagline 1 kalimat
- Screenshot hero/dashboard

2. Demo Links

- Live Demo URL
- Video Demo URL

3. About / Tentang

- Apa itu produk ini
- Siapa target user
- Masalah apa yang diselesaikan

4. Features / Fitur Utama

- List 5-6 fitur utama
- Explanation singkat per fitur

5. Screenshots

- Minimal 4 screenshots: Dashboard, Input, Alert, Detail
- Dengan caption yang jelas

6. Tech Stack

- Table framework, language, database, dll
- Justifikasi kenapa pakai tech tersebut

7. Installation / Instalasi Lokal

- Prerequisites (Node.js, npm)

- Step-by-step clone, install, setup, run
- Command yang bisa di-copy paste

8. API Endpoints (Optional)

- Table method, endpoint, description

9. Team

- Nama anggota tim + role

10. Hackathon Info

- Nama hackathon
- Tema

4.8 Bonus Points (+20 Poin)

4.8.1 Bonus 1: Advanced Tech (+10 poin)

Kriteria: Integrasi AI/ML, IoT, atau teknologi bleeding edge yang SUKSES.

Cara kita dapat:

Sistem kita mengimplementasikan Rule-Based Expert System dengan algoritma statistik:

- Multi-window EMA (Exponential Moving Average)
- Adaptive anomaly detection
- Rule-based classification engine
- Calendar-aware expected demand

Ini adalah AI dalam definisi klasik (Expert Systems) dan BUKAN sekadar CRUD!

4.8.2 Bonus 2: Deployment (+10 poin)

Kriteria: Aplikasi sudah di-deploy live (bisa diakses publik via URL).

Cara kita dapat:

Deploy ke Vercel - GRATIS dan MUDAH:

1. Push code ke GitHub
2. Ke vercel.com, login dengan GitHub
3. Import repository
4. Vercel auto-detect Next.js, klik Deploy
5. Dalam 2-3 menit, dapat URL: <https://market-pulse-radar.vercel.app>

GRATIS, HTTPS otomatis, Auto-deploy setiap push!

4.9 Penalti yang WAJIB Dihindari

WAJIB - KRITIS

DISKUALIFIKASI (Fatal):

- Video tidak bisa diputar

- Link repository private atau mati

PENALTI -20 POIN (Besar):

- Security Leak: API Key / .env ter-commit ke repo publik
- No README: Tidak ada panduan menjalankan aplikasi

PENALTI -10 POIN:

- Repository Bloat: Upload node_modules, vendor
- Spaghetti Code: File >500 baris yang campur UI, Logic, Query

4.9.1 Checklist Anti-Penalty

Check	Status	Cara Verifikasi
Video bisa diputar		Test di browser incognito
Repo tidak private		Buka di browser incognito tanpa login
Tidak ada .env di repo		Search .env di GitHub
.gitignore ada		Cek file ada di root
node_modules tidak di repo		Cek folder tidak ada di GitHub
README ada dan lengkap		Buka repo, scroll ke bawah
Tidak ada file >500 baris		Check file sizes

Part II

FORMULA MATEMATIS

5 Overview Sistem Analytics

KEY INSIGHT

Sistem analytics kita terdiri dari 6 komponen yang saling terhubung:

1. **Calendar Factors** - Multiplier berdasarkan konteks waktu
2. **Expected Demand** - Prediksi penjualan yang harusnya terjadi
3. **Momentum** - Deteksi tren naik/turun dengan EMA
4. **Burst Detection** - Deteksi anomaly/lonjakan
5. **Classification** - Kategorisasi tipe burst
6. **Priority Score** - Ranking produk

Semua formula dijelaskan secara KONSEPTUAL tanpa code implementation.

5.1 Notasi yang Digunakan

Simbol	Arti
p	Produk (identifier)
t	Waktu (hari ke-t)
$d_{p,t}$	Demand aktual produk p pada hari t
$\hat{d}_{p,t}$	Expected demand produk p pada hari t
\bar{d}_p	Baseline demand (rata-rata 30 hari)
$M_{p,t}$	Momentum score
$B_{p,t}$	Burst score
$S_{p,t}$	Priority score

6 Formula 1: Calendar Factors

6.1 Day-of-Week Factor

Konsep:

Setiap hari dalam seminggu punya pola berbeda. Weekend cenderung lebih ramai, Senin cenderung sepi.

Formula:

$$F_{dow}(t) = \frac{\bar{d}_{p,day(t)}}{\bar{d}_p}$$

Default Values (jika belum ada data cukup):

Hari	Factor	Reasoning
Senin	0.85	Awal minggu, orang masih recovery
Selasa	0.95	Mulai aktif
Rabu	1.00	Baseline (tengah minggu)
Kamis	1.05	Menjelang weekend
Jumat	1.20	TGIF, orang mulai santai
Sabtu	1.30	Weekend peak
Minggu	0.80	Banyak di rumah, family time

6.2 Payday Window Factor

Konsep:

Indonesia punya pola gajian yang khas. PNS dan swasta biasanya gajian tanggal 25 sampai awal bulan. Orang cenderung belanja lebih banyak saat fresh gajian.

Formula:

$$F_{pay}(t) = \begin{cases} 1.30 & \text{jika tanggal } \in \{25, 26, \dots, 31, 1, 2, 3, 4, 5\} \\ 0.90 & \text{jika tanggal } \in \{20, 21, 22, 23, 24\} \\ 1.00 & \text{lainnya} \end{cases}$$

Penjelasan:

- Tanggal 25-31: PNS dan sebagian swasta sudah gajian
- Tanggal 1-5: Swasta gajian, "fresh month" psychology
- Tanggal 20-24: Menjelang gajian, orang cenderung hemat

6.3 Special Day Factor

Konsep:

Hari-hari khusus punya impact besar ke penjualan. Ramadan, Lebaran, long weekend sangat mempengaruhi pola belanja.

Formula:

$$F_{special}(t) = \begin{cases} 2.00 & \text{H-7 sampai H-1 Lebaran} \\ 0.30 & \text{H+1 sampai H+7 Lebaran (kota besar)} \\ 0.50 & \text{Ramadan siang (F&B)} \\ 1.70 & \text{Ramadan malam (F&B)} \\ 1.50 & \text{Long weekend} \\ 1.00 & \text{Hari biasa} \end{cases}$$

7 Formula 2: Expected Demand

Konsep:

Expected demand adalah prediksi: "Berapa harusnya terjual hari ini?" Menggabungkan baseline dengan semua calendar factors.

Formula:

$$\hat{d}_{p,t} = \bar{d}_p \times F_{dow}(t) \times F_{pay}(t) \times F_{special}(t)$$

Contoh Perhitungan:

CONTOH**Given:**

- Produk: Nasi Goreng
- Hari: Sabtu, tanggal 27
- Baseline (rata-rata 30 hari): 50 porsi/hari

Calculation:

$$\begin{aligned}\hat{d} &= 50 \times F_{dow}(\text{Sabtu}) \times F_{pay}(27) \times F_{special} \\ &= 50 \times 1.30 \times 1.30 \times 1.00 \\ &= 84.5 \text{ porsi}\end{aligned}$$

Interpretasi:

- Jika Sabtu tanggal 27 terjual 85 porsi → **NORMAL** (sesuai ekspektasi)
- Jika terjual 150 porsi → **BURST** (perlu investigasi!)

8 Formula 3: Exponential Moving Average (EMA)

Konsep:

EMA memberikan bobot lebih besar ke data terbaru. Lebih responsif daripada simple moving average.

Formula Rekursif:

$$EMA_k(t) = \alpha \cdot d_{p,t} + (1 - \alpha) \cdot EMA_k(t - 1)$$

Smoothing Factor:

$$\alpha = \frac{2}{k + 1}$$

Nilai α untuk berbagai window:

Window (k)	α	Karakteristik
7 hari	0.25	Responsif, lebih noise
14 hari	0.133	Balanced
30 hari	0.065	Smooth, lebih stabil

Inisialisasi:

$$EMA_k(1) = d_{p,1}$$

9 Formula 4: Multi-Window Momentum

Konsep:

Momentum mengukur seberapa cepat tren berubah. Menggunakan 3 window untuk balance antara responsiveness dan stability.

Momentum untuk satu window:

$$M_{p,t}^{(k)} = \frac{EMA_k(t) - EMA_k(t-k)}{EMA_k(t-k)}$$

Kombinasi tiga window:

$$M_{p,t} = 0.5 \cdot M_{p,t}^{(7)} + 0.3 \cdot M_{p,t}^{(14)} + 0.2 \cdot M_{p,t}^{(30)}$$

Kenapa 3 window?

- **7 hari:** Menangkap pergerakan minggu ini vs minggu lalu (paling responsif)
- **14 hari:** Tren 2 mingguan
- **30 hari:** Tren bulanan (paling stabil)

Kenapa bobot 0.5, 0.3, 0.2?

- Lebih responsif ke perubahan recent (short term)
- Tapi tetap grounded oleh tren longer term

Interpretasi:

Momentum	Status	Label	Color
$M > 0.15$	Naik kuat	TRENDING UP	Green
$0.05 < M \leq 0.15$	Naik moderate	GROWING	Light Green
$-0.05 \leq M \leq 0.05$	Stabil	STABLE	Gray
$-0.15 \leq M < -0.05$	Turun moderate	DECLINING	Yellow
$M < -0.15$	Turun kuat	FALLING	Red

10 Formula 5: Adaptive Burst Score

Konsep:

Burst score mengukur seberapa "aneh" penjualan hari ini dibanding ekspektasi. Adaptive karena bandingkan dengan expected demand (context-aware), bukan cuma mean.

Formula:

$$B_{p,t} = \frac{d_{p,t} - \hat{d}_{p,t}}{\sigma_r}$$

Standard Deviation dari Residual:

$$\sigma_r = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (d_{p,t-i} - \hat{d}_{p,t-i})^2}$$

Kenapa "Adaptive"?

CONTOH

Perbandingan:

Sabtu tanggal 27, penjualan = 90 porsi, baseline = 50 porsi

Z-score biasa (SALAH):

$$z = \frac{90 - 50}{15} = 2.67 \text{ (BURST detected!)}$$

Adaptive Burst (BENAR):

$$\hat{d} = 50 \times 1.30 \times 1.30 = 84.5$$

$$B = \frac{90 - 84.5}{15} = 0.37 \text{ (Normal, sesuai ekspektasi)}$$

Adaptive burst TIDAK false positive karena paham bahwa Sabtu payday memang harusnya ramai!

Threshold:

Score	Level	Action
$B < 1.5$	Normal	Tidak perlu action
$1.5 \leq B < 2.5$	Mild	Monitor
$2.5 \leq B < 3.5$	Significant	Alert
$B \geq 3.5$	Critical	Urgent Alert

11 Formula 6: Burst Classification

Konsep:

Tidak semua burst itu sama. Viral di TikTok beda dengan naik karena weekend. Classification membantu UMKM paham akar penyebab.

Decision Tree (Pseudocode):

1. **IF** burstScore < 2.0 → **RETURN** "NORMAL"
2. **IF** hasPromo → **RETURN** "PROMO_DRIVEN"
3. **IF** (isPayday OR isWeekend):
 - **IF** burstScore < 3.0 → **RETURN** "SEASONAL"
 - **ELSE** → **RETURN** "VIRAL" (Naik melebihi seasonal expectation)
4. **IF** duration ≥ 3 hari → **RETURN** "VIRAL"
5. **IF** duration == 1 AND burstScore < 2.5 → **RETURN** "NOISE"
6. **RETURN** "MONITORING"

Penjelasan setiap kategori:

Type	Karakteristik	Rekomendasi
NORMAL	Tidak ada anomaly	Business as usual
SEASONAL	Naik sesuai pola kalender	Seharusnya sudah siap
PROMO_DRIVEN	Ada promo yang running	Akan turun setelah promo, jangan overstock
NOISE	Spike 1 hari, kecil	Ignore, random variation
MONITORING	Perlu pantau	Monitor 2-3 hari ke depan
VIRAL	Sustained, non-seasonal	Scale up! Ini opportunity

12 Formula 7: Priority Score

Konsep:

Priority score menggabungkan momentum dan burst untuk ranking produk. Produk dengan score tertinggi butuh perhatian paling besar.

Formula:

$$S_{p,t} = \lambda \cdot \tilde{M}_{p,t} + (1 - \lambda) \cdot \tilde{B}_{p,t}$$

Normalized Scores:

$$\tilde{X} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Parameter λ :

- Default: $\lambda = 0.5$ (balanced antara momentum dan burst)
- $\lambda = 0.7$: Lebih fokus ke steady growth
- $\lambda = 0.3$: Lebih sensitif ke spike/burst

Part III

PEMBAGIAN TUGAS TIM

13 Role Assignment

Person	Role	Responsibilities
A	Fullstack Lead	Arsitektur, API Routes, Database, Integrasi, Deploy
B	Frontend Dev 1	UI Components (Button, Card, Input), Styling, Charts
C	Algorithm Engineer	Semua formula di lib/analytics/, Testing formula
D	Frontend Dev 2	Pages, Layouts, User Flows, Responsive
E	PM & QA	Testing, Bug tracking, README, Video, Presentasi

14 Timeline Detail: Hari per Hari

14.1 HARI 1: Foundation & Setup

Person A (Fullstack Lead):

- Init Next.js project dengan TypeScript
- Setup Tailwind CSS
- Setup folder structure sesuai template
- Setup Vercel Postgres (atau SQLite)
- Create database schema

Person B (Frontend Dev 1):

- Setup component library structure
- Create base components: Button, Card, Input, Badge
- Setup color scheme di Tailwind config
- Create layout components

Person C (Algorithm Engineer):

- Setup lib/analytics/ folder
- Implement calendar.ts (day-of-week, payday factors)
- Implement helpers untuk date operations
- Write unit tests untuk calendar factors

Person D (Frontend Dev 2):

- Create wireframes untuk semua pages
- Setup app/ routing structure

- Create layout.tsx dengan navigation
- Create landing page (page.tsx)

Person E (PM & QA):

- Buat sample data (60 hari, 5 produk)
- Setup test scenarios document
- Create GitHub Issues untuk tracking
- Draft README structure

End of Day 1 Checkpoint:

- Next.js running locally untuk semua anggota
- Database connected
- Base components ready
- Calendar factors implemented

14.2 HARI 2: Core Backend

Person A (Fullstack Lead):

- API: POST /api/products (create product)
- API: GET /api/products (list products)
- API: POST /api/sales (input sales)
- API: GET /api/sales (list sales)
- Database queries untuk CRUD

Person B (Frontend Dev 1):

- SalesForm component (input penjualan)
- ProductCard component
- ProductList component
- Form validation

Person C (Algorithm Engineer):

- Implement EMA calculation
- Implement momentum.ts (single window)
- Write tests untuk EMA dan momentum
- Start baseline calculation

Person D (Frontend Dev 2):

- Build /input page
- Build /products page

- Connect pages ke API
- Basic navigation working

Person E (PM & QA):

- Test CRUD operations
- Document API endpoints
- Log bugs found
- Seed sample data ke database

End of Day 2 Checkpoint:

- Bisa tambah produk
- Bisa input penjualan
- Data tersimpan di database
- EMA dan momentum working

14.3 HARI 3: Core Analytics

Person A (Fullstack Lead):

- API: GET /api/analytics/momentum
- API: GET /api/analytics/burst
- Integrate analytics dengan database queries
- Optimize query performance

Person B (Frontend Dev 1):

- TrendChart component (line chart)
- AlertCard component (burst notification)
- SummaryCard component (daily summary)
- Integrate Recharts

Person C (Algorithm Engineer):

- Implement multi-window momentum (7, 14, 30)
- Implement expected demand calculation
- Implement burst.ts (adaptive burst score)
- Write comprehensive tests

Person D (Frontend Dev 2):

- Build /dashboard page
- Build /products/[id] detail page
- Integrate chart components

- Loading states

Person E (PM & QA):

- Test analytics dengan sample data
- Verify formula outputs match expected
- Document edge cases
- Prepare demo scenarios

End of Day 3 Checkpoint:

- Dashboard menampilkan summary
- Chart trend berfungsi
- Momentum dan burst score ter-hitung
- Alert muncul jika ada burst

14.4 HARI 4: Features Complete

Person A (Fullstack Lead):

- API: GET /api/analytics/ranking
- Implement priority score calculation
- Weekly report data aggregation
- Error handling untuk semua API

Person B (Frontend Dev 1):

- RankingTable component
- InsightCard component (recommendations)
- WeeklyReport component
- Badge components (TRENDING, STABLE, dll)

Person C (Algorithm Engineer):

- Implement classifier.ts (burst classification)
- Implement recommender.ts (generate recommendations)
- Implement priority.ts (priority score)
- Final testing semua formula

Person D (Frontend Dev 2):

- Build /reports page (weekly report)
- Integrate ranking ke dashboard
- Integrate recommendations ke product detail
- Navigation polish

Person E (PM & QA):

- End-to-end testing semua flow
- Test classification dengan berbagai skenario
- Bug tracking dan prioritization
- Update README dengan progress

End of Day 4 Checkpoint:

- Ranking berfungsi
- Classification benar
- Recommendations muncul
- Weekly report berfungsi
- SEMUA FITUR UTAMA SELESAI**

14.5 HARI 5: Integration & Polish

ALL: Focus pada connecting everything, fix bugs, polish

Person A (Fullstack Lead):

- Fix semua API bugs
- Optimize database queries
- Deploy ke Vercel production
- Setup environment variables di Vercel

Person B (Frontend Dev 1):

- Responsive testing (mobile, tablet, desktop)
- Fix styling issues
- Add loading skeletons
- Polish animations

Person C (Algorithm Engineer):

- Fine-tune thresholds jika perlu
- Handle edge cases (empty data, new product)
- Code cleanup
- Add comments untuk clarity

Person D (Frontend Dev 2):

- UX improvements
- Error states untuk semua pages
- Empty states

- Success/error toasts

Person E (PM & QA):

- Full regression testing
- Test di production (Vercel)
- Document semua bugs dan status
- Siapkan script untuk demo

End of Day 5 Checkpoint:

- Production deployed dan working
- Mobile responsive
- Major bugs fixed
- Happy path berjalan lancar

14.6 HARI 6: Documentation & Video Prep

Person A (Fullstack Lead):

- Final bug fixes (critical only)
- Seed production dengan demo data
- Verify deployment stable
- Help dengan screenshots

Person B (Frontend Dev 1):

- Take SEMUA screenshots untuk README
- Optimize images (compress)
- Final styling touches
- Prepare app untuk video recording

Person C (Algorithm Engineer):

- Write technical documentation
- Explain formula di README atau docs/
- Review code untuk dead code
- Remove all console.log

Person D (Frontend Dev 2):

- Final UI review
- Prepare demo flow (seed data, scenarios)
- Help dengan video recording
- Test semua links

Person E (PM & QA):

- **WRITE COMPLETE README**
- Write video script
- Setup recording environment
- Test video/audio quality

End of Day 6 Checkpoint:

- README complete dengan screenshots
- Video script finalized
- Demo data ready
- Recording environment tested

14.7 HARI 7: Video & Submit

08:00-10:00

- Final bug fixes (ONLY critical)
- **CODE FREEZE 10:00**

- Final testing di production

10:00-13:00

- **RECORD VIDEO DEMO**

- Semua anggota involved
- Multiple takes jika perlu
- Review hasil recording

13:00-16:00

- **EDIT VIDEO**

- Add intro, outro
- Add captions jika perlu
- Export final version

16:00-18:00

- Upload video ke YouTube (unlisted)
- Test playback video
- Finalize README dengan video link
- Final check semua links

18:00-20:00

- Run anti-penalty checklist
- Test repo di incognito mode
- Test Vercel URL
- **SUBMIT!**

Part IV

FINAL CHECKLIST

15 Pre-Submit Checklist

15.1 Repository

- Repo PUBLIC (cek di incognito)
- README.md ada dan lengkap
- Screenshots ada di README
- .gitignore proper
- node_modules TIDAK ada di repo
- .env TIDAK ada di repo
- .env.example ADA
- Tidak ada file > 500 baris
- Tidak ada console.log sampah

15.2 Deployment

- Vercel URL working
- HTTPS aktif
- Demo data ter-seed
- Tidak ada error di console
- Mobile responsive

15.3 Video

- Video bisa diputar (test di browser lain)
- Audio jelas
- Minimal 1080p
- Problem & Solution jelas di awal
- Demo aplikasi REAL (bukan mockup)
- Semua fitur utama ditunjukkan

15.4 Functionality

- Setup produk works
- Input penjualan works
- Dashboard shows data
- Alert muncul saat burst
- Product detail + chart works
- Ranking works
- Happy path lancar tanpa error

16 Target Score Final

Kategori	Max	Target	Strategy
Code Quality	10	10	Clean code, proper gitignore
Architecture	20	20	Clean separation, Next.js
Innovation	40	38	Expert system, context-aware
Functionality	50	48	All features working
Dok & Video	80	75	Strong storytelling
Subtotal	200	191	
Bonus: Tech	+10	+10	AI/Algorithm implemented
Bonus: Deploy	+10	+10	Vercel live
Penalty	0	0	All checks passed
TOTAL	220	211	96%

WINNING FACTOR

Dengan eksekusi yang baik dan mengikuti semua guideline di dokumen ini, target **211/220 (96%)** sangat achievable!

Kunci sukses:

1. Fokus ke VIDEO (40% bobot!) - Storytelling kuat
2. Semua FITUR JALAN - Happy path lancar
3. DEPLOY ke Vercel - Bonus +10 gratis
4. NO PENALTIES - Ikuti semua checklist