Berdasarkan kebutuhan Anda untuk 7 tim yang bekerja secara parallel namun terintegrasi, berikut rekomendasi skema terbaik:

🎯 \*\*SKEMA TERBAIK: MICROSERVICES + MONOREPO\*\*

**1. ARSITEKTUR MICROSERVICES\*\***

- Setiap tim mengembangkan service independen

- API Gateway sebagai single entry point

- Event-driven communication antar service

- Database per domain (polyglot persistence)

**2. MONOREPO DENGAN CLEAR BOUNDARIES\*\***

packages/

├── shared/ # Library bersama

├── team1-sales-mobile/

├── team2-logistics/

├── team3-finance/

├── team4-online-sales/

├── team5-offline-store/

├── team6-crm/

└── team7-hrd/

**3. TECH STACK PER TIM\*\***

- \*\*Tim 1 (Sales Mobile)\*\*: React Native + TypeScript

- \*\*Tim 2 (Logistics)\*\*: Node.js + PostgreSQL

- \*\*Tim 3 (Finance)\*\*: Java Spring Boot + MySQL

- \*\*Tim 4 (Online Sales)\*\*: Next.js + Strapi

- \*\*Tim 5 (Toko Bangunan)\*\*: Vue.js + Express.js

- \*\*Tim 6 (CRM)\*\*: Angular + .NET Core

- \*\*Tim 7 (HRD)\*\*: React + FastAPI (existing)

**4. KEUNTUNGAN UTAMA\*\***

**✅ \*\*Independent Development\*\***

- Setiap tim bisa deploy sendiri

- Technology stack sesuai kebutuhan

- Scaling independent per service

**✅ \*\*Fault Isolation\*\***

- Error di satu service tidak mematikan yang lain

- Circuit breaker pattern untuk resilience

- Bulkhead pattern untuk resource isolation

**✅ \*\*Team Autonomy\*\***

- Clear ownership per folder/service

- Independent CI/CD pipeline

- Team-specific metrics & monitoring

**✅ \*\*Integration Strategy\*\***

- API contracts yang jelas

- Event-driven communication

- Shared libraries untuk common functions

**5. WORKFLOW DEVELOPMENT\*\***

**\*\*Git Strategy:\*\***

- Feature branches per tim

- Develop branch untuk integrasi

- Release branches untuk deployment

- CODEOWNERS untuk access control

**\*\*CI/CD Pipeline:\*\***

- Per-tim testing & deployment

- Integration tests pada develop branch

- Automated security & performance testing

- Blue-green deployment untuk zero-downtime

**6. MONITORING & OBSERVABILITY\*\***

- Health check per service

- Real-time dashboard untuk semua tim

- Error tracking & alerting per tim

- Performance metrics & SLA monitoring

**7. TOOLS & INFRASTRUCTURE\*\***

- \*\*Monorepo\*\*: Lerna + Yarn Workspaces

- \*\*API Gateway\*\*: Kong/Nginx

- \*\*Message Queue\*\*: RabbitMQ

- \*\*Containers\*\*: Docker + Kubernetes

- \*\*Monitoring\*\*: Prometheus + Grafana

- \*\*CI/CD\*\*: GitHub Actions

\*\*Dengan skema ini, Anda mendapatkan:\*\*

- ✅ Parallel development yang efisien

- ✅ Fault tolerance & resilience

- ✅ Clear separation of concerns

- ✅ Scalable architecture

- ✅ Team productivity yang tinggi

Made changes.

**packages/**

├── ml-hub/ # Central ML Infrastructure

│ ├── core/ # Core ML services

│ ├── models/ # Shared models

│ ├── training/ # Training pipelines

│ └── serving/ # Model serving

│

**├── team1-sales-mobile/**

│ └── ml/ # Sales-specific ML

│ ├── recommendation-engine/ # Product recommendations

│ └── route-optimization/ # Sales route planning

│

**├── team2-logistics/**

│ └── ml/ # Logistics-specific ML

│ ├── demand-forecasting/ # Inventory prediction

│ ├── supply-chain-opt/ # Supply chain optimization

│ └── delivery-routing/ # Delivery route optimization

│

**├── team3-finance/**

│ └── ml/ # Finance-specific ML

│ ├── fraud-detection/ # Transaction fraud detection

│ ├── credit-scoring/ # Customer credit analysis

│ └── financial-forecasting/ # Cash flow prediction

│

**├── team4-online-sales/**

│ └── ml/ # E-commerce ML

│ ├── personalization/ # Product personalization

│ ├── price-optimization/ # Dynamic pricing

│ └── search-ranking/ # Search result ranking

│

**├── team6-crm/**

│ └── ml/ # CRM-specific ML

│ ├── lead-scoring/ # Lead quality scoring

│ ├── churn-prediction/ # Customer churn analysis

│ └── sentiment-analysis/ # Customer feedback analysis

│

**└── shared/**

**└── ml-common/ # Shared ML utilities**

├── data-preprocessing/

├── model-evaluation/

└── feature-engineering/

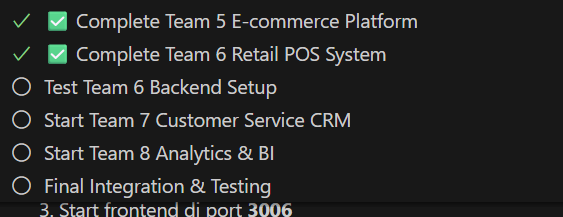
Excellent question! Untuk ML dalam arsitektur 7 tim, saya merekomendasikan \*\*hybrid approach\*\* dengan kombinasi:

Apakah Anda ingin saya:

1 Lanjutkan Team 6 POS dengan membuat POS Cashier Interface lengkap?

2 Selesaikan Team 5 dulu dengan checkout & product pages?

3 Paralel: Kerja Team 5 & Team 6 bersamaan?



## 🧠 \*\*STRATEGI ML ARCHITECTURE\*\*

### \*\*Opsi A: Centralized ML Hub + Domain-Specific Models\*\*

┌─────────────────────────────────────────────────────────────┐

│ CENTRALIZED ML HUB │

│ (Port 9000) │

├─────────────────────────────────────────────────────────────┤

│ • Core ML Infrastructure │

│ • Model Training Pipeline │

│ • Feature Store │

│ • Model Registry │

│ • A/B Testing Framework │

└─────────────────────────────────────────────────────────────┘

│

┌─────────────────────┼─────────────────────┐

│ │ │

┌───────▼──────┐ ┌────────▼──────┐ ┌────────▼──────┐

│Domain Models │ │Domain Models │ │Domain Models │

│Sales & CRM │ │Logistics & │ │Finance & │

│ │ │Inventory │ │Accounting │

└──────────────┘ └───────────────┘ └───────────────┘

```

**### \*\*REKOMENDASI: HYBRID ML ARCHITECTURE\*\* ⭐**

```

**packages/**

├── ml-hub/ # Central ML Infrastructure

│ ├── core/ # Core ML services

│ ├── models/ # Shared models

│ ├── training/ # Training pipelines

│ └── serving/ # Model serving

│

**├── team1-sales-mobile/**

│ └── ml/ # Sales-specific ML

│ ├── recommendation-engine/ # Product recommendations

│ └── route-optimization/ # Sales route planning

│

**├── team2-logistics/**

│ └── ml/ # Logistics-specific ML

│ ├── demand-forecasting/ # Inventory prediction

│ ├── supply-chain-opt/ # Supply chain optimization

│ └── delivery-routing/ # Delivery route optimization

│

**├── team3-finance/**

│ └── ml/ # Finance-specific ML

│ ├── fraud-detection/ # Transaction fraud detection

│ ├── credit-scoring/ # Customer credit analysis

│ └── financial-forecasting/ # Cash flow prediction

│

**├── team4-online-sales/**

│ └── ml/ # E-commerce ML

│ ├── personalization/ # Product personalization

│ ├── price-optimization/ # Dynamic pricing

│ └── search-ranking/ # Search result ranking

│

├── team6-crm/

│ └── ml/ # CRM-specific ML

│ ├── lead-scoring/ # Lead quality scoring

│ ├── churn-prediction/ # Customer churn analysis

│ └── sentiment-analysis/ # Customer feedback analysis

│

└── shared/

└── ml-common/ # Shared ML utilities

├── data-preprocessing/

├── model-evaluation/

└── feature-engineering/

```

---

## 🎯 \*\*ML DISTRIBUTION STRATEGY\*\*

### \*\*1. CENTRALIZED ML HUB\*\* (Team ML/Data Science)

## 🧠 \*\*STRATEGI ML TERBAIK\*\*

### \*\*1. CENTRALIZED ML HUB + DOMAIN-SPECIFIC MODELS\*\*

🏛️ ML HUB (Tim ke-8: ML/Data Science)

├── Model Registry (MLflow)

├── Feature Store

├── Training Pipeline

└── A/B Testing Framework

│

┌───────────────┼───────────────┐

│ │ │

📱 Team 1 🚛 Team 2 💰 Team 3

Sales ML Logistics ML Finance ML

├─Recommendations ├─Demand Forecast ├─Fraud Detection

├─Route Optimization ├─Inventory Opt ├─Credit Scoring

└─Performance Pred └─Delivery Pred └─Cash Flow Forecast

🌐 Team 4 🏪 Team 5 👥 Team 6

Online ML Store ML CRM ML

├─Personalization ├─POS Analytics ├─Lead Scoring

├─Dynamic Pricing ├─Store Optimization ├─Churn Prediction

└─Search Ranking └─Customer Insights └─Sentiment Analysis

```

### \*\*2. PEMBAGIAN ML PER TIM\*\*

\*\*🎯 TEAM 1 - SALES MOBILE ML:\*\*

- Product recommendation engine

- Sales route optimization

- Performance prediction

- Location-based suggestions

\*\*🚛 TEAM 2 - LOGISTICS ML:\*\*

- Demand forecasting

- Inventory optimization

- Delivery time prediction

- Supply chain optimization

\*\*💰 TEAM 3 - FINANCE ML:\*\*

- Fraud detection real-time

- Credit scoring system

- Cash flow forecasting

- Financial anomaly detection

\*\*🌐 TEAM 4 - ONLINE SALES ML:\*\*

- Product personalization

- Dynamic pricing optimization

- Search result ranking

- Customer behavior analysis

\*\*👥 TEAM 6 - CRM ML:\*\*

- Lead scoring automation

- Customer churn prediction

- Sentiment analysis

- Customer Lifetime Value

### \*\*3. KEUNTUNGAN HYBRID APPROACH\*\*

✅ \*\*Specialized Expertise\*\*: Setiap tim fokus pada domain ML mereka

✅ \*\*Shared Infrastructure\*\*: Infrastructure ML terpusat & efisien

✅ \*\*Independent Development\*\*: Tim bisa develop model secara parallel

✅ \*\*Standardized Operations\*\*: Consistent ML ops across teams

✅ \*\*Easy Scaling\*\*: Model bisa di-scale independent per kebutuhan

### \*\*4. ML INFRASTRUCTURE\*\*

- \*\*ML Hub\*\*: MLflow + Feature Store + Model Serving (Port 9000-9005)

- \*\*Shared Libraries\*\*: Common ML utilities untuk semua tim

- \*\*Model Registry\*\*: Centralized model versioning & deployment

- \*\*A/B Testing\*\*: Framework untuk test model performance

- \*\*Monitoring\*\*: Real-time model performance tracking

\*\*Dengan strategi ini:\*\*

1. \*\*Tim ML/Data Science\*\* handle infrastructure & advanced ML

2. \*\*Setiap business tim\*\* develop domain-specific models

3. \*\*Shared components\*\* ensure consistency & efficiency

4. \*\*Independent deployment\*\* per tim tanpa dependencies

Apakah Anda setuju dengan strategi ML ini? Atau ingin saya lanjutkan dengan implementasi setup lengkap untuk semua tim termasuk ML infrastructure?

Starting (1/8) \*Setup Monorepo Structure\*

Made changes.