Integration Status — AISignalEngine

TL;DR

- **Sudah masuk (built-in):** engine core, patterns, Kelly-lite sizing, OpenAI fallback, backtest dummy, genetic loop (manual start), logging, neutral-signal safety.
- **Belum masuk (but stub-ready):** real **Technical/LuxAlgo feed**, real **CVD/OrderBook feed**, **performance tracker** (persist outcome → update win_rate/pattern confidence secara dinamis).

---

1) Real Data Hooks — Status

Komponen	Status	Di Kode	Catatan Implementasi
Technical Service (RSI/ EMA/ LuxAlgo)	**Belum**	`TechnicalData` masih dummy	Tambahkan service & inject → isi `trend_direction`, `trend_strength`, RSI real
CVD/ OrderBook (OB)	**Belum**	`gatherComprehensiveMarketData()` dummy	Wire ke CVD, orderbook imbalance, liquidity pools, POC/VP
Funding Enhanced (Coinglass/ OKX)	**Ada**	`enhancedFundingRateService`	Sudah dipakai untuk pattern funding squeeze

Contoh Stub (TypeScript)

```
export interface TechnicalService {
  getSnapshot(symbol: string): Promise<{
    rsi: number; ema20: number; ema50: number;
    trend: "bullish"|"bearish"|"neutral"|"strong_bullish"|"strong_bearish";
    strength: number; // 0..1
  }>;
}

export interface OrderflowService {
  getCVD(symbol: string): Promise<{ ratio: number; dominant:
"buy"|"sell"|"balanced" }>;
  getOrderbookImbalance(symbol: string): Promise<{ bidAskImb: number;
largeWalls: number }>; // normalized
}

constructor(deps: Dependencies = {}) {
  // ...
```

```
  this.deps.technicalService = deps.technicalService; // optional
  this.deps.orderflowService = deps.orderflowService; // optional
}
```

---

2) Performance Tracker — Status

Fitur	Status	Di Kode	Catatan
Simpan Outcome Signal	**Belum** (logging only)	`storeSignalForLearning()`	Perlu tabel/collection `signals` dan `trades`
Update Dinamis win_rate/pattern confidence	**Belum**	—	Hitung dari hasil eksekusi & mark-to-market

Skema Minimal (DB)

```
-- signals (generate)
signal_id TEXT PK,
symbol TEXT,
ts TIMESTAMP,
direction TEXT,
confidence REAL,
rr REAL

-- executions (result)
signal_id TEXT FK,
entry_price REAL,
exit_price REAL,
size REAL,
return REAL,
duration_hours REAL
```

Hook Update (Pseudo)

```
async recordExecution(result: ExecutionResult) {
  await db.insert(result);
  const stats = await db.calcStats({ horizonDays: 30 });
  this.dynamicWinRate = stats.winRate;
  this.patterns.set("whale_accumulation", {
    ...p,
    historical_accuracy: stats.winRate,
  });
}
```

---

3) Quick TODO Checklist

- [ ] **TechnicalService**: tarik RSI/EMA/LuxAlgo → isi `TechnicalData`
- [ ] **OrderflowService**: CVD + OB imbalance → `gatherComprehensiveMarketData()`
- [ ] **Performance Tracker**: tabel `signals` / `executions` + updater win_rate & confidence
- [ ] **Telegram Notifier**: human+JSON message saat signal & saat TP/SL hit
- [ ] **Executor Adapter**: OKX/Bybit untuk konversi SL/TP (decimal) → harga

---

4) Integration Hints

- SL/TP **decimal** → konversi harga di executor: `priceTP = entry * (1 ± tp)`
- Sizing pakai **Kelly-lite** (sudah di engine) → jaga max 30% notional
- Jalankan `startSchedulers()` hanya di worker khusus, bukan di web thread