

# Kandungan

|   |                                                                |    |
|---|----------------------------------------------------------------|----|
| • | Tentang Pengarang .....                                        | 6  |
| • | Penafian Risiko Kewangan .....                                 | 7  |
| • | Penafian Pengaturcaraan dan Keselamatan .....                  | 8  |
| • | Amaran Pelaksanaan Automatik .....                             | 9  |
| • | Penafian Maklumat Syariah .....                                | 10 |
| • | Notis Regulatori Malaysia .....                                | 11 |
| • | Prakata Pengarang .....                                        | 13 |
| • | Rekap Siri ezyKripto: Buku 01 hingga 04 .....                  | 15 |
| • | Cara Menggunakan Buku Ini .....                                | 17 |
| • | Prasyarat dan Apa yang Tidak Diandaikan .....                  | 19 |
| • | Persediaan Teknologi .....                                     | 20 |
| • | Peta Pembelajaran .....                                        | 22 |
| • | Pendedahan Kepentingan Pengarang .....                         | 24 |
| • | Dari Satu Trading Idea kepada Sistem yang Berfungsi 24/7 ..... | 28 |

## BAHAGIAN I • ASAS ALGORITHMIC TRADING

|    |                                                      |    |
|----|------------------------------------------------------|----|
| 01 | Daripada Manual Trading ke Algorithmic Trading ..... | 34 |
| 02 | Anatomi Sebuah Trading Bot .....                     | 42 |
| 03 | Python untuk Crypto Trading .....                    | 51 |
| 04 | Bina Paper Trading Bot Pertama Anda .....            | 62 |

## BAHAGIAN II • MARKET DATA, EXCHANGE API & DATA ENGINEERING

|    |                                                     |     |
|----|-----------------------------------------------------|-----|
| 05 | REST API, WebSocket & Exchange Connectivity .....   | 72  |
| 06 | CCXT – Satu Interface untuk Pelbagai Exchange ..... | 84  |
| 07 | Membina Crypto Market Data Pipeline .....           | 93  |
| 08 | Real-Time Data & Event-Driven Architecture .....    | 105 |

## BAHAGIAN III • STRATEGY ENGINEERING & BACKTESTING

|    |                                                    |     |
|----|----------------------------------------------------|-----|
| 09 | Menukarkan Trading Idea kepada Rules .....         | 115 |
| 10 | Backtesting dengan Python .....                    | 126 |
| 11 | Freqtrade & Hummingbot Masterclass .....           | 136 |
| 12 | Statistical Validation & Strategy Robustness ..... | 147 |

**BAHAGIAN IV • RISK ENGINEERING & EXECUTION SYSTEMS**

|           |                                                    |            |
|-----------|----------------------------------------------------|------------|
| <b>13</b> | <b>Position Sizing &amp; Portfolio Risk Engine</b> | <b>160</b> |
| <b>14</b> | <b>Order Management &amp; Execution Engine</b>     | <b>175</b> |
| <b>15</b> | <b>Paper Trading vs Live Execution</b>             | <b>188</b> |
| <b>16</b> | <b>Bina Paper Execution Engine Profesional</b>     | <b>198</b> |

**BAHAGIAN V • VPS, AUTOMATION & OPERATIONS**

|           |                                                   |            |
|-----------|---------------------------------------------------|------------|
| <b>17</b> | <b>Ubuntu VPS untuk Trading Systems</b>           | <b>210</b> |
| <b>18</b> | <b>Docker, Database &amp; Service Deployment</b>  | <b>220</b> |
| <b>19</b> | <b>n8n Trading Automation</b>                     | <b>233</b> |
| <b>20</b> | <b>Telegram, Discord &amp; Trading Monitoring</b> | <b>245</b> |

**BAHAGIAN VI • HERMES & AGENTIC TRADING SYSTEMS**

|           |                                                      |            |
|-----------|------------------------------------------------------|------------|
| <b>21</b> | <b>Apa Itu AI Agent dalam Trading?</b>               | <b>257</b> |
| <b>22</b> | <b>Hermes AI Agent Setup &amp; Integration</b>       | <b>266</b> |
| <b>23</b> | <b>Multi-Agent Crypto Trading Architecture</b>       | <b>278</b> |
| <b>24</b> | <b>Agent Prompts, Tools, Memory &amp; Guardrails</b> | <b>292</b> |

**BAHAGIAN VII • PROJEK CAPSTONE LENGKAP**

|           |                                                                  |            |
|-----------|------------------------------------------------------------------|------------|
| <b>25</b> | <b>Projek A – Bot Paper Trading Spot Lengkap</b>                 | <b>305</b> |
| <b>26</b> | <b>Projek B – On-Chain Intelligence ke Paper Signal Pipeline</b> | <b>315</b> |
| <b>27</b> | <b>Projek C – Meja Penyelidikan Crypto Agentic</b>               | <b>325</b> |
| <b>28</b> | <b>Projek D – VPS Deployment &amp; Operational Resilience</b>    | <b>337</b> |

**BAHAGIAN VIII • RISIKO, TADBIR URUS & PENGUASAAN AKHIR**

|           |                                                                               |            |
|-----------|-------------------------------------------------------------------------------|------------|
| <b>29</b> | <b>Trading System Cybersecurity</b>                                           | <b>349</b> |
| <b>30</b> | <b>Algorithmic &amp; Agentic Trading dari Perspektif Syariah dan Regulasi</b> | <b>362</b> |
| <b>31</b> | <b>Sepuluh Kajian Kes Kegagalan Realistik</b>                                 | <b>372</b> |
| <b>32</b> | <b>Dari Trader kepada System Architect – Capstone 30 Hari</b>                 | <b>382</b> |

**LAMPIRAN & RUJUKAN**

|          |                                                  |            |
|----------|--------------------------------------------------|------------|
| <b>A</b> | <b>Glosari Algorithmic &amp; Agentic Trading</b> | <b>395</b> |
| <b>•</b> | <b>Buku Kerja Praktikal – 22 Worksheet</b>       | <b>410</b> |
| <b>•</b> | <b>Penilaian Akhir &amp; Skema Jawapan</b>       | <b>444</b> |
| <b>B</b> | <b>Rujukan Pantas Python</b>                     | <b>474</b> |
| <b>C</b> | <b>Rujukan Pantas CCXT (Baca Sahaja)</b>         | <b>481</b> |
| <b>D</b> | <b>Konsep REST &amp; WebSocket</b>               | <b>488</b> |
| <b>E</b> | <b>Templat Strategi</b>                          | <b>495</b> |

|          |                                                     |            |
|----------|-----------------------------------------------------|------------|
| <b>F</b> | Senarai Semak Backtesting .....                     | <b>500</b> |
| <b>G</b> | Formula Risiko & Contoh Pengiraan .....             | <b>504</b> |
| <b>H</b> | Seni Bina Sistem Paper Trading .....                | <b>510</b> |
| <b>I</b> | Rujukan Pantas Freqtrade .....                      | <b>520</b> |
| <b>J</b> | Rujukan Pantas Hummingbot .....                     | <b>526</b> |
| <b>K</b> | Rujukan Hermes Agent dan Meja Penyelidikan .....    | <b>530</b> |
| <b>L</b> | Rujukan Workflow n8n .....                          | <b>536</b> |
| <b>M</b> | Rujukan Telegram dan Discord .....                  | <b>541</b> |
| <b>N</b> | Arahan Ubuntu dan Docker .....                      | <b>545</b> |
| <b>O</b> | Senarai Semak Keselamatan .....                     | <b>551</b> |
| <b>P</b> | Playbook Kegagalan Sistem .....                     | <b>555</b> |
| <b>Q</b> | Senarai Semak Penilaian Syariah .....               | <b>562</b> |
| <b>R</b> | Rujukan Regulatori Malaysia .....                   | <b>566</b> |
| <b>S</b> | Panduan Repositori Kod Penuh .....                  | <b>571</b> |
| <b>T</b> | Penjejak Capstone 30 Hari .....                     | <b>583</b> |
| <b>U</b> | Soalan Lazim (FAQ) .....                            | <b>587</b> |
| <b>V</b> | Dokumentasi Rasmi dan Bacaan Lanjut .....           | <b>595</b> |
| <b>W</b> | Retrospektif Lengkap Siri ezyKripto Lima Buku ..... | <b>604</b> |
| <b>•</b> | Rujukan & Sumber Disahkan .....                     | <b>610</b> |

BAHAGIAN I

# Asas Algorithmic Trading

Daripada trading manual kepada sistem berasaskan rules: anatomi sebuah trading bot, Python untuk trader, dan bot paper trading pertama anda.



**BAB 01** Daripada Manual Trading ke Algorithmic Trading

**BAB 02** Anatomi Sebuah Trading Bot

**BAB 03** Python untuk Crypto Trading

**BAB 04** Bina Paper Trading Bot Pertama Anda

# Daripada Manual Trading ke Algorithmic Trading

Automasi tidak mencipta edge — ia melaksanakan proses dengan lebih konsisten.

Jam loceng telefon berbunyi pada 3:55 pagi di sebuah apartmen di Cheras. Pemiliknya, seorang akauntan yang berdagang BTC/USDT pada carta 4 jam selepas waktu pejabat, bangun untuk melihat candle yang ditutup tepat pada 04:00 MYT. Peraturannya mudah: beli apabila purata bergerak pendek melintas ke atas purata bergerak panjang, dan keluar apabila ia melintas semula ke bawah. Malangnya, pada minggu yang sama dia terlepas dua candle penting kerana tertidur dan satu lagi kerana mesyuarat yang berlarutan.

Keesokan harinya sebuah iklan muncul dalam suapan media sosialnya: "Bot AI patuh Syariah, untung harian, tanpa perlu memantau carta." Tawaran itu menggoda kerana masalahnya nyata. Namun soalan yang lebih penting ialah: apa sebenarnya yang ingin diautomasikan? Peraturan yang sudah diuji, atau harapan bahawa mesin akan menemui keuntungan sendiri?

Bab ini membina peta konsep untuk seluruh buku: lapan cara trading dilakukan, dan di mana keputusan dibuat, dilaksanakan dan dikawal.

## © OBJEKTIF PEMBELAJARAN

- |                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>01</b> Membezakan manual, rule-based, algorithmic, automated, AI-assisted, agentic dan multi-agent trading, serta peranan deterministic execution.</p> <p><b>02</b> Menerangkan maksud edge dan kenapa automasi tidak menciptanya.</p> <p><b>03</b> Mengenal pasti lapisan seni bina sebuah sistem trading profesional.</p> | <p><b>04</b> Menghuraikan pipeline daripada penyelidikan hingga reconciliation.</p> <p><b>05</b> Menjelaskan kenapa output LLM bukan sistem algoritma yang telah disahkan.</p> <p><b>06</b> Menilai sama ada sesuatu proses trading sudah bersedia untuk diautomasikan.</p> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## 01.1 Apa yang sebenarnya berubah apabila trading diautomasikan

Setiap dagangan, sama ada dibuat dengan jari atau dengan kod, melalui empat soalan yang sama: apakah keadaan pasaran sekarang, apakah keputusan yang patut diambil, berapa banyak risiko yang dibenarkan, dan bagaimana order dilaksanakan. Dalam trading manual, keempat-empatnya dijawab oleh orang yang sama, sering dalam beberapa saat dan dipengaruhi emosi. Automasi tidak menghapuskan soalan ini; ia memindahkan jawapan ke dalam rules yang ditulis, diuji dan dijalankan oleh komputer.

Manfaat utamanya ialah konsistensi. Komputer tidak tertidur pada 04:00 MYT dan tidak "memberi peluang sekali lagi" kepada dagangan yang rugi. Tetapi konsistensi berfungsi ke dua arah. Jika rules itu tidak mempunyai

kelebihan statistik, komputer akan melaksanakan kerugian itu dengan konsisten juga — lebih cepat, lebih kerap dan tanpa rasa ragu.

Di sinilah konsep **edge** menjadi penting. Edge ialah kelebihan yang boleh diukur, iaitu jangkaan pulangan positif selepas kos, yang kekal apabila diuji pada data yang tidak digunakan untuk mereka bentuk strategi. Edge datang daripada penyelidikan yang jujur, bukan daripada bahasa pengaturcaraan, kelajuan VPS atau label "AI". Pengajaran teras buku ini wajar diingat sejak awal: sistem trading yang baik tidak semestinya sistem yang autonomi. Sistem automasi yang boleh dipercayai memerlukan penyelidikan yang sah, pelaksanaan terkawal, risiko yang boleh diukur dan disiplin operasi.

◇ CONTOH NUMERIK

**Berapa banyak titik keputusan yang terlepas?**

Carta 4 jam menghasilkan 6 candle sehari. Sempadan candle (UTC) 00:00, 04:00, 08:00, 12:00, 16:00 dan 20:00 bersamaan dengan 08:00, 12:00, 16:00, 20:00, 00:00 dan 04:00 MYT.

- 1. Andaikan trader hanya boleh menyemak carta antara 07:00 dan 23:00 MYT.
- 2. Candle yang ditutup dalam tempoh itu: 08:00, 12:00, 16:00, 20:00 MYT = 4 daripada 6.
- 3. Titik keputusan yang terlepas:  $2 \div 6 \approx 33.3\%$  setiap hari, atau  $2 \times 365 = 730$  penutupan candle setahun.

Automasi boleh menutup jurang ini. Tetapi ia juga bermakna setiap kos dagangan dibayar pada setiap signal. Pada data sintetik buku, strategi SMA 20/50 menghasilkan +2.99% tanpa kos tetapi hanya +0.29% selepas yuran, spread dan slippage (Bab 10). Konsistensi melaksanakan segala-galanya, termasuk kos.

**01.2 Lapan kategori: daripada manual hingga multi-agent**

Istilah seperti "bot", "algo" dan "AI trading" sering digunakan secara longgar. Buku ini menggunakan lapan kategori berlabel A hingga H, dibezakan oleh tiga soalan: siapa yang membuat keputusan, siapa yang melaksanakan order, dan apa yang mengawal kedua-duanya.

| KOD | KATEGORI                | SIAPA MEMUTUSKAN                       | SIAPA MELAKSANA                          | KAWALAN UTAMA                   |
|-----|-------------------------|----------------------------------------|------------------------------------------|---------------------------------|
| A   | Manual (discretionary)  | Trader, berdasarkan pertimbangan       | Trader                                   | Disiplin peribadi               |
| B   | Rule-based              | Rules bertulis, dibaca oleh trader     | Trader                                   | Senarai semak                   |
| C   | Algorithmic             | Kod mengira signal daripada rules      | Trader atau sistem                       | Ujian kod, backtest             |
| D   | Automated               | Kod                                    | Sistem, tanpa campur tangan setiap order | Risk engine, kill switch        |
| E   | AI-assisted             | Manusia atau kod, dibantu ringkasan AI | Manusia atau kod                         | Semakan manusia                 |
| F   | Agentic                 | AI agent merancang dan memanggil alat  | Alat yang dibenarkan sahaja              | Kebenaran alat, pengesah        |
| G   | Multi-agent             | Beberapa agent dengan peranan berbeza  | Perkhidmatan berasingan                  | Orchestrator, pemisahan peranan |
| H   | Deterministic execution | Tiada keputusan baharu                 | Enjin pelaksanaan                        | Input sama → output sama        |

**Rajah 1.1** menunjukkan spektrum yang sama secara visual: semakin ke kanan, semakin banyak tugas dipindahkan daripada manusia kepada perisian, dan semakin penting kawalan yang tidak boleh dipintas.

## SPEKTRUM A-H • SIAPA MEMUTUS, SIAPA MELAKSANA, APA YANG MENGAWAL

| Kategori                         | Siapa membuat keputusan                           | Siapa melaksana                                | Kawalan utama                                 |
|----------------------------------|---------------------------------------------------|------------------------------------------------|-----------------------------------------------|
| <b>A</b> Manual trading          | Trader, secara discretionary                      | Trader menekan butang                          | Disiplin peribadi, jurnal                     |
| <b>B</b> Rule-based trading      | Peraturan bertulis; manusia mengikutnya           | Trader                                         | Checklist & playbook                          |
| <b>C</b> Algorithmic trading     | Kod (rules dalam algoritma)                       | Manusia atau sistem                            | Backtest & pengesahan statistik               |
| <b>D</b> Automated trading       | Kod, tanpa campur tangan                          | Sistem automatik 24/7                          | Risk engine, kill switch, pemantauan          |
| <b>E</b> AI-assisted trading     | Manusia; AI meringkas & mencadang                 | Manusia atau sistem deterministik              | Semakan manusia, pengesah output              |
| <b>F</b> Agentic trading         | AI agent merancang & memanggil alat               | <b>Dalam buku ini: tiada akses pelaksanaan</b> | Kebenaran alat, guardrails                    |
| <b>G</b> Multi-agent trading     | Beberapa agent + orchestrator (nasihat)           | <b>Tiada — output ialah laporan</b>            | Pengasingan peranan, skema JSON               |
| <b>H</b> Deterministic execution | Tiada keputusan baharu; ikut keputusan yang lulus | Perkhidmatan paper deterministik               | RiskEngine fail-closed, audit, reconciliation |

■ Manusia memutus (A–B)  
 ■ Kod memutus (C–D)  
 ■ AI menyelidik & menasihati (E–G)  
 ■ Kawalan deterministik (H)

**RAJAH 1.1 — Manual vs Algorithmic vs Agentic Trading.** Kategori trading berbeza pada siapa yang memutus dan siapa yang melaksana; dalam buku ini agent AI hanya menyelidik, manakala pelaksanaan kekal deterministik.

EZK5-002

## Bertindih, bukan tangga

Sebuah sistem boleh berada dalam beberapa kategori serentak. Strategi algorithmic (C) boleh dilaksanakan secara manual: komputer menghantar signal, trader menekan butang sendiri. Meja penyelidikan agentic (F) boleh membantu sistem rule-based tanpa pernah menyentuh order.

Dua istilah lagi sering muncul dan bertindih dengan jadual di atas. **Systematic trading** bermaksud rules yang sama digunakan secara konsisten pada setiap peluang; ia meliputi B, C dan D. **Quantitative trading** pula menekankan asal-usul rules: ia diterbitkan daripada analisis statistik atau model matematik, bukan sekadar pemerhatian carta. Strategi quantitative hampir selalu algorithmic, tetapi tidak sebaliknya: crossover dua purata bergerak ialah rules algorithmic yang ringkas, bukan model kuantitatif.

Kategori H berbeza sifatnya. **Deterministic execution** ialah syarat kejuruteraan, bukan gaya trading: input yang sama mesti menghasilkan keputusan yang sama, setiap kali. Dalam buku ini, kategori D hingga G mesti duduk di atas H — agent boleh mencadangkan, tetapi saiz, semakan risiko dan kebenaran order dikira oleh kod deterministik.

### ◆ TERMINOLOGI

**Manual (discretionary) trading** — keputusan dan pelaksanaan oleh trader berdasarkan pertimbangan semasa.

**Rule-based trading** — trader mengikut rules bertulis yang spesifik dan melaksanakan sendiri.

**Algorithmic trading** — rules ditulis sebagai kod yang mengira signal secara automatik daripada data.

**Systematic trading** — rules yang sama digunakan secara konsisten pada setiap peluang.

**Quantitative trading** — rules diterbitkan dan diuji melalui kaedah statistik atau matematik.

**Automated trading** — sistem menghantar order tanpa kelulusan manusia bagi setiap order, di bawah had tetap.

**AI-assisted trading** — AI membantu analisis (ringkasan, klasifikasi); keputusan kekal pada manusia atau kod deterministik.

**Agentic trading** — AI agent merancang langkah dan memanggil alat secara berulang untuk mencapai matlamat.

**Multi-agent trading** — beberapa AI agent dengan peranan berasingan diselaraskan oleh orchestrator.

**Deterministic** — input yang sama sentiasa menghasilkan output yang sama; boleh diuji dan diulang.

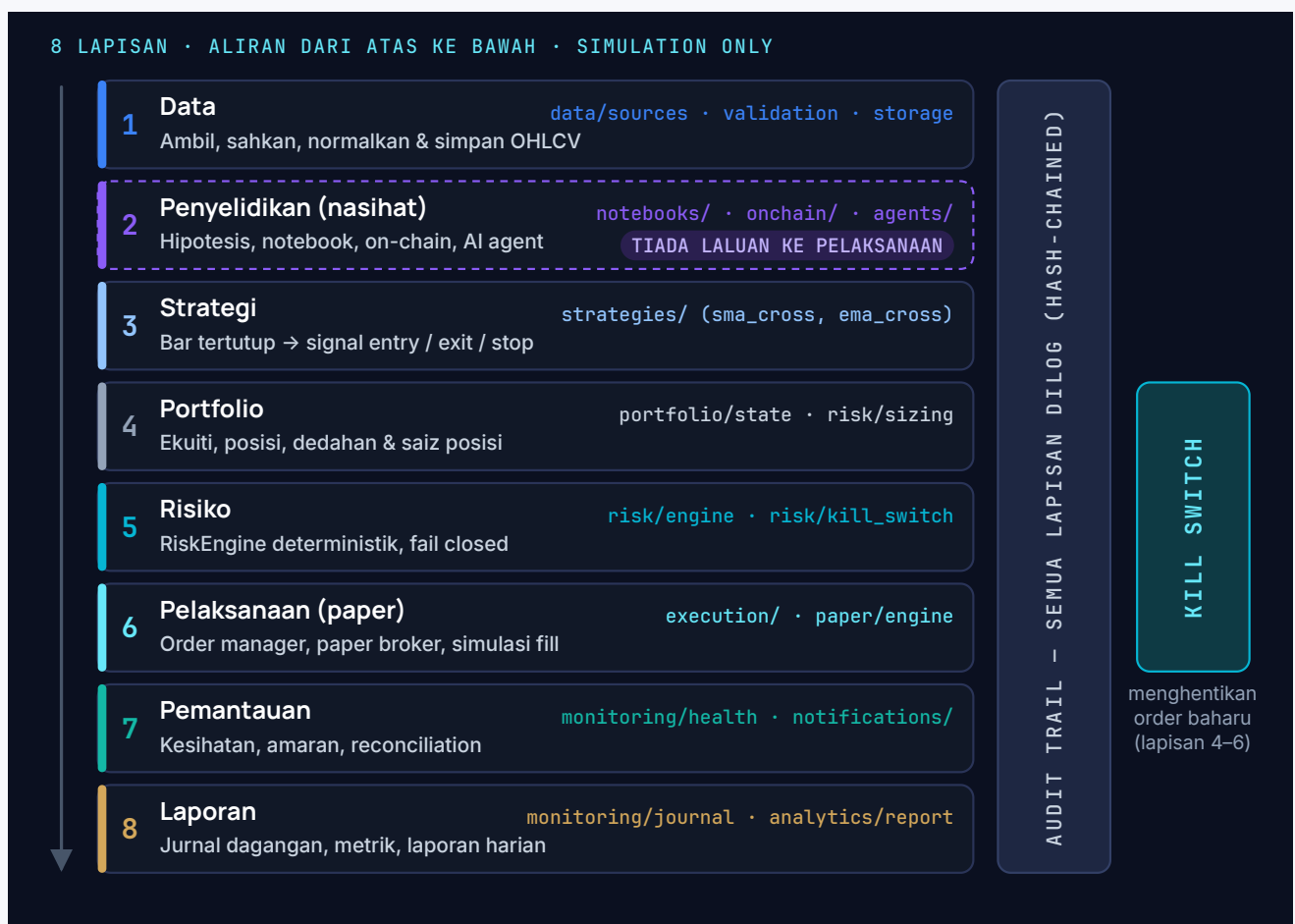
**Edge** — jangkaan pulangan positif selepas kos yang kekal pada data yang tidak digunakan semasa reka bentuk.

**Human oversight** — keupayaan manusia untuk memantau, menghentikan dan menyemak sistem, dengan kuasa yang tidak boleh dipintas oleh perisian.

### 01.3 Seni bina sistem trading profesional

Dalam kepala manusia, membaca carta, menilai risiko dan menekan butang berlaku dalam satu aliran fikiran. Dalam perisian, setiap tugas itu menjadi lapisan tersendiri dengan input, output dan cara gagal yang tersendiri.

**Rajah 1.2** menyusun lapan lapisan: **data** (kumpul dan sahkan harga), **penyelidikan** (uji idea), **strategi** (signal), **portfolio** (padankan dengan pegangan), **risiko** (semak had), **pelaksanaan** (order dan fill), **pemantauan** (kesihatan lapisan lain) dan **laporan** (rekod untuk manusia).



**RAJAH 1.2 — Seni Bina Sistem Trading Profesional.** Sistem trading profesional berlapis: data mengalir ke bawah melalui strategi, portfolio dan RiskEngine sebelum pelaksanaan paper, sementara audit trail merekod semua lapisan.

EZK5-003

Pemisahan ini menentukan apa yang berlaku apabila sesuatu rosak. Jika data basi, lapisan risiko menolak order; jika pelaksanaan menerima timeout, reconciliation menyemak keadaan sebenar sebelum apa-apa dihantar semula. Bab 02 memecahkan lapisan ini kepada komponen terperinci.

## 01.4 Pipeline daripada penyelidikan hingga reconciliation

Lapisan menerangkan *struktur*; pipeline menerangkan *urutan*. **Rajah 1.3** menunjukkan enam peringkat yang dilalui setiap idea sebelum ia menjadi kedudukan dalam akaun paper.

SATU ARAH • SETIAP PERINGKAT MEMPUNYAI OUTPUT YANG BOLEH DIPERIKSA



**RAJAH 1.3 — Pipeline Penyelidikan ke Pelaksanaan.** Idea bergerak sehalu daripada penyelidikan ke signal, keputusan portfolio, pengesahan risiko, pelaksanaan paper dan reconciliation; hanya peringkat pertama boleh dibantu AI.

EZK5-004

- Research** — idea diuji pada data sejarah dengan kos realistik; outputnya spesifikasi dan bukti, bukan janji.
- Signal** — pada setiap candle yang telah ditutup, strategi mengira sama ada syarat entry atau exit dipenuhi.
- Portfolio decision** — adakah sudah ada posisi, atau dedahan kepada aset berkorelasi sudah tinggi?
- Risk validation** — risk engine menyemak saiz, had kerugian, kesegaran data, mod sistem dan kill switch. Satu kegagalan bermakna order ditolak.
- Execution** — order dihantar kepada broker, dalam buku ini sentiasa paper broker atau mock exchange.
- Reconciliation** — rekod dalaman dibandingkan dengan venue; percanggahan menggantung order baharu.

Signal hanya satu daripada enam peringkat. Trader baharu menghabiskan hampir semua masa pada peringkat kedua, sedangkan kebanyakan kegagalan sistem berlaku pada peringkat ketiga hingga keenam.

Peringkat kedua itu sendiri boleh ditulis sebagai function yang deterministik. Contoh di bawah ialah ilustrasi ringkas rules crossover; ia dipanggil tiga kali dengan input yang sama dan sekali dengan input lain.

```
Ilustrasi: rules crossover yang deterministik
```

```

1 def sma_signal(prev_fast, prev_slow, fast, slow):
2     if prev_fast <= prev_slow and fast > slow:
3         return "BELI pada open bar seterusnya"
4     if prev_fast >= prev_slow and fast < slow:
5         return "JUAL pada open bar seterusnya"
6     return "TIADA TINDAKAN"
7
8
9 for _ in range(3):
10     print(sma_signal(49_900, 50_000, 50_050, 50_010))
11     print(sma_signal(50_050, 50_010, 50_080, 50_020))

```

PYTHON

```

Output
BELI pada open bar seterusnya
BELI pada open bar seterusnya
BELI pada open bar seterusnya
TIADA TINDAKAN

```

Input yang sama memberi jawapan yang sama. Sifat inilah yang membolehkan rules diuji dengan unit test, dibandingkan antara backtest dan paper trading, dan diaudit selepas kejadian. Frasa "pada open bar seterusnya" pula mencerminkan peraturan masa yang dilaksanakan dalam Bab 04: signal dikira selepas candle ditutup.

Companion code buku ini menguatkuasakan satu lagi kawalan pada tahap reka bentuk. Mod dagangan ditakrifkan sebagai senarai tertutup, dan tiada pilihan untuk wang sebenar:

```

companion-code/src/ezykripto/config.py · baris 20-23
PYTHON
20 class TradingMode(str, Enum):
21     BACKTEST = "BACKTEST" # historical replay
22     PAPER = "PAPER"       # simulated fills against (replayed or public) prices
23     SHADOW = "SHADOW"     # decisions are logged, nothing is filled

```

Tiada ahli LIVE dalam TradingMode, dan load\_settings() menolak sebarang konfigurasi yang meminta live execution, walaupun API key dimasukkan. Sistem pembelajaran tidak sepatutnya mempunyai suis tersembunyi ke akaun sebenar.

## 01.5 LLM bukan sistem algoritma yang telah disahkan

Model bahasa besar (LLM) menimbulkan godaan baharu: kenapa menulis rules jika kita boleh bertanya kepada AI sama ada hendak beli atau jual? Rules crossover di atas deterministik, boleh diuji pada ribuan candle sejarah dan memberi jawapan yang sama esok. LLM pula menjana teks secara probabilistik. Soalan yang sama boleh menghasilkan jawapan berbeza, model boleh dikemas kini tanpa notis, dan jawapan yang yakin boleh mengandungi fakta yang direka (hallucination).

AI tetap berguna untuk meringkaskan berita, mengklasifikasikan pengumuman atau menyemak log, asalkan outputnya disemak. Tetapi apabila tugas itu menentukan kuantiti, membenarkan order atau mematikan kill switch, kita memerlukan kod yang boleh diuji baris demi baris. AI agent boleh membantu menganalisis keadaan pasaran, tetapi keputusan untuk membenarkan order mestilah melalui risk engine deterministik yang tidak boleh dipintas oleh arahan bahasa semula jadi. Bahagian VI buku ini membina seni bina agentic di atas prinsip tersebut.

### ↔ FAKTA VS MITOS

**Mitos:** Bot berlabel "AI" lebih pintar, jadi ia tidak memerlukan backtest atau had risiko.

**Fakta:** Label tidak mengubah keperluan pengesahan. Output AI ialah input yang tidak dipercayai sehingga disahkan, dan had risiko mesti dikuatkuasakan oleh kod yang deterministik.

## 01.6 Bila automasi masuk akal, dan bila belum

Kembali kepada akauntan di Cheras. Sebelum membeli atau membina apa-apa bot, dia perlu melalui empat ujian kesediaan.

Pertama, **penyelidikan yang sah**: adakah rules itu cukup spesifik sehingga dua orang akan bertindak sama pada candle yang sama, dan sudahkah ia diuji dengan kos pada data baharu? Kedua, **pelaksanaan terkawal**: apa yang berlaku jika exchange tidak menjawab atau signal yang sama diterima dua kali? Ketiga, **risiko yang boleh diukur**: adakah saiz dikira daripada had kerugian, dan adakah had harian yang menghentikan sistem? Keempat,

**disiplin operasi:** siapa memantau, bagaimana dia dimaklumkan, dan bagaimana sistem dihentikan dengan selamat?

Jika mana-mana jawapan "belum pasti", langkah seterusnya ialah paper trading dengan rekod lengkap — laluan buku ini daripada rules bertulis kepada kod, backtest dan paper execution yang dipantau. Langkah ke wang sebenar memerlukan semakan berasingan di luar skop buku ini.

#### ◆ KONTEKS MALAYSIA

- **Masa:** sistem dalam buku ini menyimpan masa dalam UTC dan memaparkannya dalam MYT (UTC+8).
- **Menjual strategi automatik:** pada 2018, SC menyatakan bahawa penyediaan "automated trading and investment strategy" kepada pelabur ialah aktiviti dikawal selia, dan mengeluarkan arahan kepada Dagang Algomecs Sdn Bhd untuk berhenti [REG-018]. Ini berkaitan dengan menawarkan strategi kepada orang lain; Bab 30 membincangkannya.
- **Skim berjenama bot:** Senarai Amaran Pelabur SC memuatkan beberapa skim yang menggunakan jenama bot atau AI, termasuk yang meletakkan perkataan "Halal" pada nama (setakat Oktober 2026) [REG-018]. Semak status entiti sebelum berurusan.
- **Tanggungjawab API:** terma penggunaan Luno dan Hata meletakkan tanggungjawab setiap permintaan API pada pengguna, dan exchange boleh membatalkan akses API [REG-033, REG-034].

#### ► DALAM SIRI EZYKRIPTO

Buku 02 (Perpetual Futures Trading) mengajar position sizing manual, expectancy dan pengesahan playbook strategi dengan tangan; Bab 20 buku itu menunjuk ke buku ini untuk backtesting automatik. Buku 05 mengambil disiplin yang sama dan menjadikannya kod yang boleh diuji.

#### ► LATIHAN LANGKAH DEMI LANGKAH

##### Klasifikasikan lima situasi

1. Tentukan kategori (A–H) bagi setiap situasi; satu situasi boleh mempunyai lebih daripada satu kod.
2. (a) Trader membuka carta, menilai "rasa" pasaran dan membeli. (b) Skrip Python menghantar mesej Telegram apabila SMA20 melintas SMA50; trader memutuskan sendiri. (c) Sistem paper menghantar order simulasi setiap kali signal muncul, tertakluk kepada risk engine. (d) Chatbot meringkaskan berita pagi; trader membaca sebelum membuat keputusan. (e) Tiga AI agent menyediakan laporan pasaran, on-chain dan risiko yang disatukan oleh orchestrator, tanpa akses kepada order.
3. Untuk setiap situasi, tulis siapa yang membuat keputusan dan siapa yang melaksanakan.

**JAWAPAN:** (a) A. (b) C (signal algorithmic, pelaksanaan manual). (c) D di atas H. (d) E. (e) G (dan F bagi setiap agent); kerana tiada akses order, sistem ini kekal sebagai alat penyelidikan.

#### × KESILAPAN IMPLEMENTASI

- **Mengautomasikan rules yang belum ditulis.** "Beli bila momentum kuat" tidak boleh diuji. Akibat: kod membuat tafsiran yang tidak disedari trader. Kawalan: tulis spesifikasi strategi lengkap dahulu (Bab 09).
- **Menyamakan backtest dengan bukti edge.** Akibat: strategi yang terlalu disesuaikan gagal pada data baharu. Kawalan: pengesahan out-of-sample dan walk-forward (Bab 12).
- **Memberi AI kuasa ke atas saiz atau order.** Akibat: output probabilistik menentukan wang. Kawalan: AI hanya menasihati; risk engine deterministik membuat keputusan.
- **Tiada cara menghentikan sistem.** Akibat: kesilapan berulang tanpa had. Kawalan: kill switch yang boleh diaktifkan oleh sesiapa tetapi hanya dinyahaktifkan oleh operator (Bab 13).

#### ▲ AMARAN RISIKO

Automasi menggandakan kelajuan, bukan kebijaksanaan: sistem yang salah boleh menghantar puluhan order dalam beberapa minit. Semua latihan dalam buku ini berjalan dalam PAPER MODE dengan data sintetik atau mock exchange; jangan sambungkan kod buku ini kepada akaun yang mempunyai dana sebenar.

## ● RINGKASAN BAB

**5 Perkara Utama**

1. Automasi memindahkan keputusan ke dalam rules bertulis; ia tidak mencipta edge.
2. Lapan kategori A–H dibezakan oleh siapa memutuskan, siapa melaksana dan apa yang mengawal.
3. Sistem trading profesional dibina daripada lapisan data, penyelidikan, strategi, portfolio, risiko, pelaksanaan, pemantauan dan laporan.
4. Signal hanya satu daripada enam peringkat pipeline; kebanyakan kegagalan berlaku selepasnya.
5. LLM boleh membantu analisis, tetapi had risiko dan kebenaran order mesti dikuatkuasakan oleh kod deterministik.

**5 Soalan Ulang Kaji**

1. Apakah beza antara algorithmic trading dan automated trading?
2. Kenapa systematic dan quantitative trading dikatakan bertindih dengan kategori lain?
3. Apakah yang dimaksudkan dengan deterministic execution, dan kenapa ia menjadi asas kategori D–G?
4. Namakan enam peringkat pipeline daripada research hingga reconciliation.
5. Apakah empat ujian kesediaan sebelum sesuatu proses trading diautomasikan?

## ✓ SENARAI SEMAK PENGESAHAN

- ☐ Saya boleh mengklasifikasikan sesuatu sistem ke dalam kategori A–H dan menerangkan sebabnya.
- ☐ Saya boleh mendefinisikan edge tanpa menyebut keuntungan yang dijanjikan.
- ☐ Saya boleh melukis lapan lapisan seni bina daripada ingatan.
- ☐ Saya faham kenapa TradingMode dalam companion code tidak mempunyai mod LIVE.
- ☐ Saya boleh menerangkan kenapa output LLM perlu disahkan sebelum digunakan.

## → SETERUSNYA

Sekarang kita mempunyai peta kategori dan lapisan. Bab 02 membuka "kotak hitam" sebuah trading bot dan memecahkannya kepada komponen yang boleh diuji secara berasingan — daripada data feed hingga audit trail — serta memetakan setiap komponen kepada folder dalam pakej ezykripto.

# Anatomi Sebuah Trading Bot

Bot yang boleh dipercayai ialah himpunan komponen kecil yang setiap satunya tahu bila untuk berkata tidak.

Nurul (watak rekaan), seorang jurutera perisian di Cyberjaya, menerima mesej daripada rakan lamanya pada suatu malam Jumaat. Rakannya telah menulis `bot.py` — satu fail sepanjang 600 baris yang memuat turun harga, mengira purata bergerak, menentukan saiz, menghantar order, menyimpan log dan menghantar notifikasi Telegram. Bot itu "berfungsi" selama dua minggu dalam mod ujian. Kemudian satu permintaan ke exchange tamat masa, skrip mencuba semula secara automatik, dan log menunjukkan dua order beli untuk signal yang sama. Tiada siapa yang tahu dengan pasti order mana yang benar-benar wujud.

Nurul tidak terus membaca kod. Dia mengambil pen penanda dan melukis kotak di papan putih: satu kotak untuk data, satu untuk strategi, satu untuk risiko, satu untuk order, satu untuk rekod. Kemudian dia bertanya soalan yang tidak pernah ditanya oleh skrip itu: "Kotak mana yang sepatutnya berkata *tidak*?"

Bab ini mengikuti lukisan Nurul: memecahkan bot kepada komponen dengan satu tanggungjawab, mengikuti aliran data hingga laporan, dan memetakan setiap komponen kepada folder dalam companion code.

## © OBJEKTIF PEMBELAJARAN

- |                                                                                                                                                                                                                                                                        |                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>01</b> Menamakan komponen utama sebuah trading bot dan tanggungjawab setiap satu.</p> <p><b>02</b> Menerangkan aliran MARKET DATA hingga REPORTING untuk satu candle.</p> <p><b>03</b> Menjelaskan kenapa strategi, risiko dan pelaksanaan mesti dipisahkan.</p> | <p><b>04</b> Membezakan infrastruktur penyelidikan dengan infrastruktur pelaksanaan.</p> <p><b>05</b> Mencari setiap komponen dalam pakej ezykripto/.</p> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|

## 02.1 Kenapa satu skrip besar berbahaya

Skrip monolitik seperti `bot.py` cepat ditulis dan boleh dibaca dari atas ke bawah. Masalahnya muncul apabila sesuatu gagal. Jika fungsi yang menghantar order juga mengira saiz dan juga mencuba semula selepas timeout, tiada sempadan yang jelas di mana keputusan "jangan hantar lagi" boleh dibuat. Setiap bahagian bergantung pada keadaan dalaman bahagian lain.

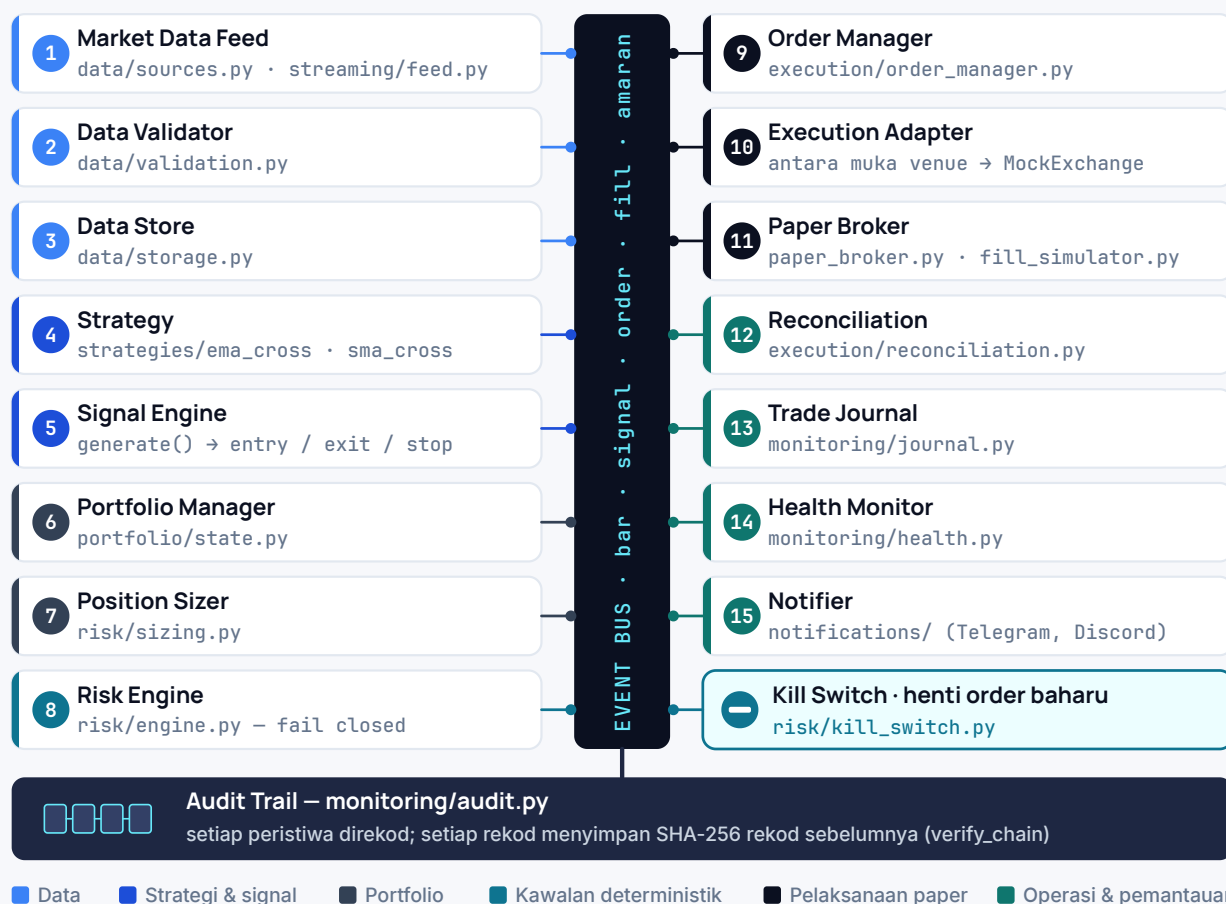
Gandingan yang ketat ini merosakkan tiga perkara. Pertama, **ujian**: anda tidak boleh menguji logik risiko tanpa juga menjalankan kod yang memanggil exchange. Kedua, **pengasingan kegagalan**: ralat kecil dalam format notifikasi boleh menghentikan seluruh gelung trading, atau lebih buruk, membiarkannya berjalan tanpa rekod. Ketiga, **akauntabiliti**: apabila sesuatu yang salah berlaku, log tidak menunjukkan komponen mana yang membuat keputusan.

Dunia kewangan tradisional mempunyai contoh yang mahal tentang betapa pentingnya kawalan operasi di sekeliling kod. Pada 1 Ogos 2012, Knight Capital mengedarkan kod baharu yang tidak disalin ke satu daripada lapan pelayannya; dalam kira-kira 45 minit sistem itu menghasilkan lebih empat juta pelaksanaan dan kerugian melebihi \$460 juta <sup>[INC-001, INC-002]</sup>. Kita tidak membina sistem sebesar itu, tetapi pengajarannya sama: perisian trading gagal pada sempadan antara komponen — penempatan, konfigurasi, pengesahan — bukan hanya dalam formula strategi.

## 02.2 Peta komponen

**Rajah 2.1** menunjukkan peta komponen yang akan kita bina sepanjang buku. Setiap kotak mempunyai satu tanggungjawab, dan komponen bertukar mesej berstruktur — candle, signal, permintaan order, keputusan risiko, fill — yang boleh direkod dan diulang.

15 komponen • event bus • kill switch • audit trail — modul dalam `src/ezykripto/`



**RAJAH 2.1 — Peta Komponen Trading Bot.** Lima belas komponen berkomunikasi melalui event bus; kill switch menghentikan order baharu dan audit trail merekod setiap peristiwa dengan rantaian hash.

EZK5-005

Berikut ialah komponen utama, dikumpulkan mengikut fungsi:

- **Masukan:** *data feed* (sumber harga, sama ada fail, REST atau WebSocket), *data validation* (menolak candle yang rosak, berganda atau basi) dan *konfigurasi* (parameter dan rahsia, dibaca sekali semasa permulaan).
- **Keputusan:** *strategy* (rules yang menukar candle tertutup kepada signal), *signal engine* (yang menjalankan strategi pada setiap candle baharu), *portfolio manager* (yang menyemak pegangan semasa) dan *risk engine* (yang meluluskan atau menolak setiap order).
- **Pelaksanaan:** *order manager* (kitaran hayat order, client order ID dan pencegahan order berganda), *execution adapter* (lapisan yang bercakap dengan venue), *paper broker* dan *fill simulator* (yang meniru fill tanpa wang sebenar), serta *portfolio state* (tunai, posisi, ekuiti).

- **Kawalan dan rekod:** *reconciliation* (membandingkan rekod dalaman dengan venue), *monitoring* (kesihatan setiap komponen), *audit trail* (rekod yang tidak boleh diubah tanpa dikesan), *notifikasi* dan *reporting* (jurnal dan laporan harian).

Di antara komponen ini terdapat dua elemen merentas. **Event bus** ialah saluran yang membawa mesej dari satu komponen ke komponen seterusnya. Dalam sistem besar ia mungkin message queue berasingan; dalam companion code, ia ialah urutan panggilan yang tetap dalam satu proses dan sebuah queue terhad untuk data streaming (Bab 08). **Kill switch** pula ialah suis yang, apabila aktif, menghentikan semua order automatik baharu, tidak kira apa yang dikatakan oleh strategi.

#### ◆ TERMINOLOGI

**Data feed** — sumber data pasaran (fail, REST, WebSocket) yang membekalkan candle atau tick kepada sistem.

**Signal engine** — komponen yang menjalankan strategi pada setiap data baharu dan menerbitkan signal.

**Strategy** — rules yang menukar candle tertutup kepada signal entry, exit dan stop; tidak mengetahui baki akaun.

**Portfolio manager** — komponen yang menilai signal berdasarkan pegangan, dedahan dan korelasi semasa.

**Risk engine** — komponen deterministik yang meluluskan atau menolak setiap order berdasarkan had tetap; gagal secara tertutup.

**Order manager** — komponen yang mengurus kitaran hayat order, ID unik dan pencegahan order berganda.

**Execution adapter** — lapisan yang menterjemah permintaan order kepada format venue tertentu.

**Paper broker** — broker simulasi yang menerima order dan menghasilkan fill tanpa wang sebenar.

**Event bus** — saluran mesej berstruktur antara komponen, supaya setiap langkah boleh direkod dan diulang.

**Audit trail** — rekod kronologi keputusan dan tindakan yang membolehkan pengubahan dikesan.

**Kill switch** — mekanisme yang menghentikan semua order automatik baharu dengan serta-merta.

**Research vs execution infrastructure** — pemisahan antara persekitaran percubaan (notebook, backtest, agent) dan perkhidmatan terkawal yang mengendalikan order.

### 02.3 Aliran satu candle: daripada data ke laporan

Untuk memahami bagaimana komponen bekerjasama, ikuti satu candle BTC/USDT 4 jam yang ditutup pada 20:00 MYT. [Rajah 2.2](#) menunjukkan sepuluh peringkat yang dilaluinya.

## 10 LANGKAH • SETIAP LANGKAH BOLEH MENGHENTIKAN ALIRAN



**RAJAH 2.2 — Aliran Data ke Laporan.** Setiap bar melalui sepuluh langkah; validasi data, RiskEngine dan reconciliation masing-masing boleh menghentikan aliran sebelum kerosakan berlaku.

EZK5-103

- MARKET DATA** — candle yang dibuka pada 08:00 UTC (16:00 MYT) dan ditutup pada 12:00 UTC (20:00 MYT) tiba daripada data feed. Masa disimpan dalam UTC, dan setiap candle dilabel dengan masa bukannya.
- DATA VALIDATION** — sistem menyemak: adakah  $\text{high} \geq \text{open}$  dan  $\text{close}$ , adakah volum tidak negatif, adakah candle sebelumnya wujud, adakah data cukup segar? Jika tidak, aliran berhenti di sini.
- STRATEGY** — strategi mengira penunjuk daripada candle tertutup sahaja.
- SIGNAL** — strategi menerbitkan, contohnya, "entry long pada open bar seterusnya, stop pada harga X".
- PORTFOLIO CHECK** — adakah sudah ada posisi BTC? Adakah dedahan kepada kumpulan aset utama sudah penuh?
- RISK ENGINE** — saiz dikira daripada risk budget; semua had disemak; kill switch dan kesihatan sistem disemak. Satu kegagalan bermakna penolakan.
- PAPER EXECUTION** — order manager memberi client order ID, paper broker mengisi order pada open candle seterusnya (12:00 UTC, 20:00 MYT) — harga pertama yang wujud selepas keputusan — dengan yuran dan slippage simulasi.
- RECONCILIATION** — posisi dalaman dibandingkan dengan rekod broker.
- MONITORING** — setiap komponen melaporkan kesihatan; sebarang kegagalan menghentikan order baharu.
- REPORTING** — fill direkod dalam jurnal; ringkasan dihantar kepada operator.

Perhatikan bahawa empat peringkat — DATA VALIDATION, PORTFOLIO CHECK, RISK ENGINE dan RECONCILIATION — wujud terutamanya untuk berkata *tidak*. Inilah jawapan kepada soalan Nurul. Dalam bot.py, tiada satu pun kotak itu wujud secara berasingan, jadi tiada tempat untuk menolak percubaan kedua.

#### ◇ CONTOH NUMERIK

##### Kos satu order berganda

Gunakan contoh sizing kanonik buku (Bab 13): ekuiti 10,000 USDT, risiko 0.5% = 50 USDT, entry 50,000, stop 49,500, yuran 0.10% dan slippage 5 bps setiap arah.

- Saiz yang betul: **0.07701 BTC**, notional **3,850.50 USDT**, rugi pada stop  $\approx$  **50.00 USDT**.
- Jika timeout menyebabkan order dihantar semula tanpa semakan, posisi menjadi  $2 \times 0.07701 =$  **0.15402 BTC**.
- Notional berganda:  $2 \times 3,850.50 =$  **7,701.00 USDT**.
- Rugi pada stop kira-kira  $2 \times 50.00 =$  **100.00 USDT**, iaitu 1% ekuiti — dua kali ganda had yang ditetapkan.

Tiada formula strategi yang salah dalam contoh ini. Kegagalan berlaku sepenuhnya dalam lapisan pelaksanaan, dan hanya order manager serta reconciliation yang boleh mencegahnya (Bab 14).

## 02.4 Kontrak antara komponen

Pemisahan hanya bermakna jika setiap komponen mematuhi kontrak yang jelas. Kontrak paling penting dalam buku ini ialah antara strategi dan bahagian lain sistem. Docstring antara muka strategi dalam companion code menyatakannya terus:

```
companion-code/src/ezykripto/strategies/base.py PYTHON
1 """Strategy interface.
2
3 A strategy turns CLOSED bars into signals. It never sizes positions, never
4 talks to a broker and never sees account balances: those belong to the risk
5 engine and the order manager. Column contract of ``generate``:
6
7 * ``entry`` bool – open a long at the NEXT bar's open
8 * ``exit`` bool – close the long at the NEXT bar's open
9 * ``stop`` float – protective stop price for a new entry (NaN if none)
10 """
```

Strategi hanya melihat candle tertutup dan menerbitkan tiga lajur: entry, exit dan stop. Kontrak yang mengehadkan ini membolehkan strategi yang sama dijalankan dalam backtest (Bab 10), enjin paper (Bab 16) dan ujian unit, dengan keputusan yang boleh dibandingkan trade demi trade.

Kontrak kedua ialah sifat risk engine. Prinsipnya ditulis di bahagian atas modulnya:

```
companion-code/src/ezykripto/risk/engine.py PYTHON
1 """Deterministic pre-trade RiskEngine (Bab 13; requirements in research/risk-engine-spec.md).
2
3 Principles
4 1. FAIL CLOSED. Every check must positively pass. Missing information, an
5   exception inside a check, or an unknown value is a rejection.
6 2. All checks run and are all reported, so an operator sees every reason, but a
7   single failure rejects the order.
8 3. Nothing outside this module can loosen a limit at run time. ``RiskLimits`` is
9   frozen; there is no "override" argument; AI agents have no handle to it.
10 4. Orders whose ``source`` is ``agent``, ``chat`` or ``workflow`` are rejected:
11   language models, chat messages and automation tools may suggest research,
12   they may not originate orders.
13 5. Risk-REDUCING orders (reduce-only sells of an existing long) are still allowed
14   when loss limits are breached, so a position can be closed – but not when the
15   kill switch is active, data is stale or the system is unhealthy.
16 """
```

Prinsip pertama — *fail closed* — bermakna maklumat yang hilang atau ralat dalam semakan dianggap sebagai penolakan, bukan kelulusan. Prinsip keempat menolak order yang berasal daripada agent, chat atau workflow. Kontrak ketiga ialah kill switch: sesiapa boleh mengaktifkannya, tetapi hanya operator boleh menyahaktifkannya, dengan sebab bertulis:

```

companion-code/src/ezykripto/risk/kill_switch.py PYTHON
1 """Kill switch.
2
3 * Anyone (human, automated rule, monitoring) may ACTIVATE it.
4 * Only an actor with role ``operator`` may DEACTIVATE it, and must give a reason.
5   AI agents, chat users and workflows are never operators.
6 * State lives in a file so it survives restarts and can be set from a shell
7   (``touch var/paper/KILL_SWITCH``) even if every Python service is broken.
8 """

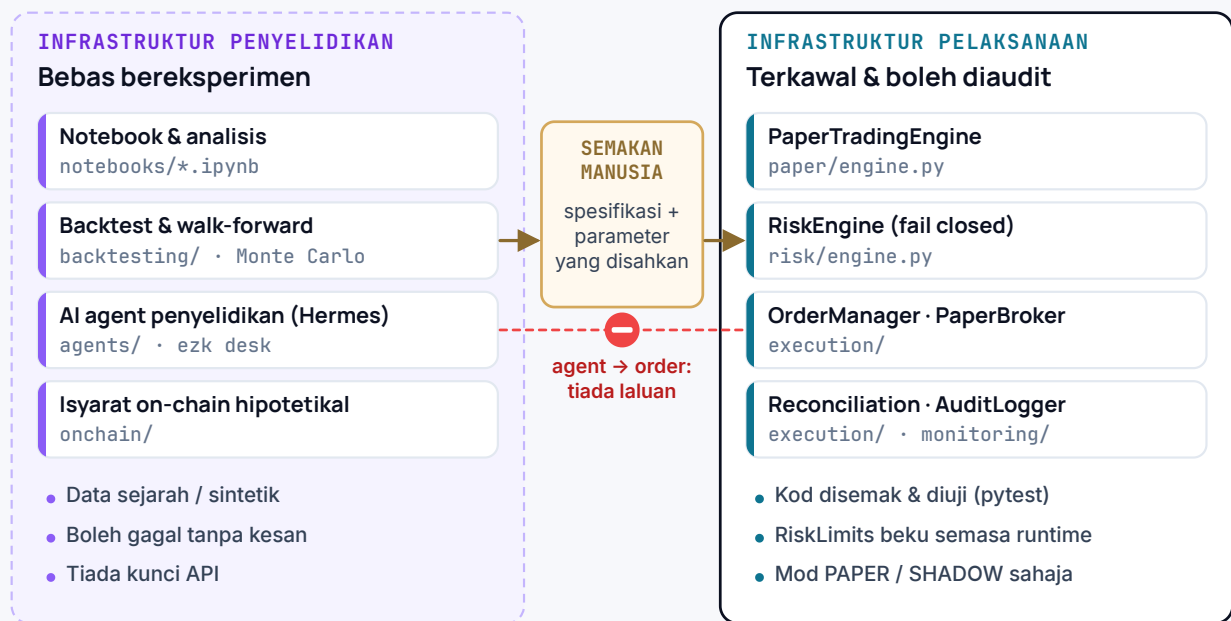
```

Ketiga-tiga kontrak ini mempunyai satu tema: komponen yang paling dekat dengan wang mempunyai kuasa paling sedikit untuk dilonggarkan oleh komponen lain.

## 02.5 Infrastruktur penyelidikan berbanding infrastruktur pelaksanaan

Satu lagi garis yang dilukis Nurul di papan putih memisahkan dua dunia. Di sebelah kiri ialah **infrastruktur penyelidikan**: notebook, skrip backtest, eksperimen parameter dan, kemudian dalam buku ini, AI agent yang menyediakan laporan. Di sebelah kanan ialah **infrastruktur pelaksanaan**: perkhidmatan yang berjalan berterusan, menerima data, menyemak risiko dan menghasilkan fill paper. [Rajah 2.3](#) menggambarkan kedua-dua zon ini.

### DUA ZON · SATU PINTU YANG DIKAWAL



Perubahan bergerak dari kiri ke kanan hanya melalui semakan, ujian dan deployment yang direkod.

**RAJAH 2.3 — Infrastruktur Penyelidikan vs Pelaksanaan.** Penyelidikan boleh gagal tanpa kesan; perubahan masuk ke infrastruktur pelaksanaan hanya melalui semakan manusia, ujian dan deployment yang direkod.

EZK5-104

Keperluan kedua-dua zon bertentangan. Penyelidikan memerlukan kebebasan: cuba sepuluh idea, buang sembilan. Pelaksanaan memerlukan kestabilan: kod disemak, versi dipin, log lengkap, perubahan yang perlahan. Jika kedua-duanya bercampur — contohnya, notebook yang sama digunakan untuk eksperimen dan menghantar order — satu eksperimen yang terlupa boleh menjadi order.

## NOTA SENI BINA

**Kenapa risk engine berada dalam proses yang sama dengan enjin paper**

Dalam companion code, risk engine dipanggil terus oleh enjin paper, bukan melalui rangkaian. Trade-off: kita kehilangan keupayaan untuk menskalakan risiko secara berasingan, tetapi kita memperoleh jaminan bahawa tiada order boleh sampai ke broker tanpa melalui semakan risiko, walaupun rangkaian dalaman gagal. Bagi sistem pembelajaran, jaminan itu lebih bernilai. Bab 18 membincangkan keputusan ini semula dalam konteks Docker.

Pemisahan ini juga menentukan aliran maklumat. Penyelidikan menghasilkan *spesifikasi* dan *parameter* yang disemak, kemudian dipindahkan ke zon pelaksanaan secara sengaja. Zon pelaksanaan menghasilkan *rekod* — fill, jurnal, log audit — yang dibaca kembali oleh penyelidikan. Tiada laluan langsung daripada AI agent di zon kiri ke order di zon kanan; Bab 23 dan 24 menunjukkan bagaimana sempadan ini dikuatkuasakan dalam kod.

**02.6 Peta pakej ezykripto/**

Companion code buku ini mengikut peta komponen di atas. Jadual berikut menunjukkan folder utama dalam companion-code/src/ezykripto/ dan bab yang membincangkannya.

| FOLDER               | KOMPONEN                                                    | BAB UTAMA  |
|----------------------|-------------------------------------------------------------|------------|
| config.py, errors.py | Konfigurasi, mod dagangan, ralat                            | 03, 15, 29 |
| data/                | Data feed, pengesahan, storan, resampling                   | 07         |
| streaming/           | Pemproses event, queue terhad, resync                       | 08         |
| strategies/          | Strategy, penunjuk, SMA & EMA crossover                     | 04, 09     |
| backtesting/         | Enjin backtest, metrik, walk-forward, Monte Carlo           | 10, 12     |
| risk/                | Risk engine, sizing, kill switch                            | 13         |
| execution/           | Order manager, paper broker, fill simulator, reconciliation | 14, 16     |
| portfolio/           | Portfolio state (tunai, posisi, ekuiti)                     | 16         |
| monitoring/          | Audit trail, kesihatan, jurnal                              | 16, 28     |
| paper/               | Enjin paper trading, drill kegagalan                        | 16, 28     |
| notifications/       | Format mesej, Telegram, Discord, HMAC webhook               | 19, 20     |
| agents/              | Skema, pengesah, kebenaran alat, meja penyelidikan          | 21–24, 27  |
| onchain/             | Pipeline isyarat on-chain hipotetikal                       | 26         |
| services/            | API status tempatan, pelayan MCP baca-sahaja                | 18, 22     |

Antara muka baris arahan ezk mengumpulkan komponen ini kepada lapan projek. Docstring di bahagian atas cli.py berfungsi sebagai indeks:

```
companion-code/src/ezykripto/cli.py PYTHON
1 """Command-line entry point: `ezk <command>` (or `python -m ezykripto <command>`).
2
3 Every command is simulation-only. Projects in the book map to commands:
4 P01 Market Data Collector ..... ezk sample-data | ezk validate | ezk collect
5 P02 Strategy Backtesting Engine ..... ezk backtest | ezk walkforward | ezk montecarlo | ezk scorecard
6 P03 Risk Engine ..... ezk risk-demo
7 P04 Paper Execution Engine ..... ezk paper | ezk paper-service | ezk drills
8 P05 Trading Journal & Analytics ..... ezk journal
9 P06 n8n Notification Workflows ..... ezk notify-test | ezk sign-webhook
10 P07 Hermes Read-Only Research Agents . ezk research-tools | ezk mcp-server
11 P08 Integrated Multi-Agent Desk ..... ezk desk
12 """
```

Setiap arahan berjalan dalam mod simulasi; anda akan memasang pakej dalam Bab 03.

◆ KONTEKS MALAYSIA

- **Penanda aras amalan baik:** Garis Panduan Pengurusan Risiko Teknologi SC (SC-GL/2-2023, disemak 2024) mengikat entiti berlesen, bukan individu. Namun prinsip AI/ML di dalamnya — termasuk pelan kontingensi yang boleh "promptly suspend AI applications", keupayaan menerangkan keputusan algoritma dan rekod untuk kebolehsesanan — ialah penanda aras berguna bagi kill switch dan audit trail anda [REG-027].
- **Akauntabiliti API:** terma Luno dan Hata menyatakan pengguna bertanggungjawab atas setiap permintaan API, sama ada dibenarkan atau tidak [REG-033, REG-034]. Audit trail yang lengkap ialah cara anda mengetahui apa yang sebenarnya dihantar.
- **Masa dalam log:** simpan UTC dalam pangkalan data dan log; paparkan MYT dalam notifikasi kepada operator di Malaysia.

► LATIHAN LANGKAH DEMI LANGKAH

Petakan skrip monolitik kepada komponen

1. Senaraikan tugas bot.py rakan Nurul: muat turun harga, kira SMA, tentukan saiz, hantar order, cuba semula selepas timeout, tulis log, hantar Telegram.
2. Untuk setiap tugas, tulis komponen yang sepatutnya memilikinya berdasarkan [Rajah 2.1](#).
3. Kenal pasti tugas yang tiada dalam skrip tetapi diperlukan.
4. Buka folder companion-code/src/ezykripto/ dan cari folder bagi setiap komponen yang anda tulis.

**JAWAPAN:** muat turun harga → data feed (data/); kira SMA → strategy (strategies/); saiz → risk engine (risk/sizing.py); hantar order → order manager dan execution adapter (execution/); cuba semula → order manager, dengan status UNKNOWN dan semakan venue, bukan hantar semula membuta tuli; log → audit trail (monitoring/audit.py); Telegram → notifikasi (notifications/). Yang tiada: data validation, portfolio check, reconciliation, kill switch dan monitoring.

× KESILAPAN IMPLEMENTASI

- **Strategi yang mengira saiz sendiri.** Akibat: had risiko boleh dilangkau apabila strategi diubah. Kawalan: strategi hanya menerbitkan signal; sizing dalam risk engine.
- **Cuba semula automatik pada lapisan pelaksanaan.** Akibat: order berganda selepas timeout. Kawalan: client order ID deterministik dan semakan status sebelum menghantar semula (Bab 14).
- **Log hanya ke skrin.** Akibat: tiada rekod selepas proses dimulakan semula. Kawalan: audit trail berasaskan fail dengan rantaian hash (Bab 16).
- **Notebook penyelidikan yang boleh menghantar order.** Akibat: eksperimen menjadi dagangan. Kawalan: zon penyelidikan tidak mempunyai kelayakan atau laluan kod ke broker.

**🛡️ NOTA KESELAMATAN**

Komponen yang memegang rahsia (API key, token bot) mesti sesedikit mungkin dan diasingkan daripada komponen yang menerima input luar seperti mesej chat atau output AI. Dalam buku ini, tiada komponen memegang kunci dengan kebenaran trade atau withdrawal; Bab 05 dan Bab 29 menerangkan hierarki kebenaran dan pengurusan rahsia.

**● RINGKASAN BAB****5 Perkara Utama**

1. Skrip monolitik sukar diuji, tidak mengasingkan kegagalan dan tidak menunjukkan siapa membuat keputusan.
2. Bot yang baik terdiri daripada komponen dengan satu tanggungjawab dan antara muka berstruktur.
3. Empat peringkat dalam aliran — pengesahan data, semakan portfolio, risk engine, reconciliation — wujud untuk menolak.
4. Strategi tidak mengetahui baki akaun; risk engine gagal secara tertutup; kill switch hanya dinyahaktifkan oleh operator.
5. Infrastruktur penyelidikan dan pelaksanaan dipisahkan; tiada laluan langsung daripada eksperimen atau AI ke order.

**5 Soalan Ulang Kaji**

1. Apakah tiga perkara yang dirosakkan oleh gandingan ketat dalam skrip monolitik?
2. Susun sepuluh peringkat aliran daripada MARKET DATA hingga REPORTING.
3. Kenapa strategi dalam companion code tidak dibenarkan menentukan saiz posisi?
4. Apakah maksud "fail closed" dalam konteks risk engine?
5. Apakah perbezaan keperluan antara infrastruktur penyelidikan dan infrastruktur pelaksanaan?

**✓ SENARAI SEMAK PENGESAHAN**

- ☐ Saya boleh menamakan sekurang-kurangnya dua belas komponen dan tanggungjawabnya.
- ☐ Saya boleh menerangkan peringkat mana yang menolak order dan sebabnya.
- ☐ Saya telah membaca docstring `strategies/base.py`, `risk/engine.py` dan `risk/kill_switch.py`.
- ☐ Saya boleh mencari folder bagi setiap komponen dalam `src/ezykripto/`.
- ☐ Saya faham kenapa zon penyelidikan tidak mempunyai laluan ke order.

**→ SETERUSNYA**

Peta komponen memberitahu kita *apa* yang perlu dibina. Bab 03 menyediakan alatnya: Python, virtual environment, pandas, pytest dan konsep pengaturcaraan asas yang digunakan oleh setiap komponen dalam pakej ezykripto, melalui siri skrip kecil yang boleh anda jalankan sendiri.