# Blueprint Perbaikan Sistem AutoAnalisa (Spot & Futures Pullback 1–4%)

## 1. Arsitektur Modul (Pemisahan SPOT vs FUTURES, Payload vs Rules)

Untuk meningkatkan modularitas, arsitektur sistem AutoAnalisa akan dipisahkan jelas antara logika **Spot** dan **Futures**, serta antara modul **payload builder** (pengumpulan data) dan **rules engine** (deteksi sinyal). Pemisahan ini mengikuti panduan pada dokumen-dokumen sebelumnya:

* **Modul Data (Datasource)**: Data pasar Spot dan Futures ditangani oleh modul terpisah. Contohnya: autoanalisa/datasource/binance\_spot.py untuk data Spot dan autoanalisa/datasource/binance\_futures.py untuk Futures[[1]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=3.1%20%60binance_spot.py%60%20Fungsi%3A%20,tidak%2C%20gunakan%20default%20dari%20config). Masing-masing modul menyediakan fungsi-fungsi REST sync untuk mengambil **exchange info**, **klines** multi-timeframe, dan depth orderbook sesuai market. Pemisahan ini memastikan perbedaan data antara Spot dan Futures tertangani di layer data source, **sebelum** agregasi di level berikutnya[[2]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=,round_to%28x%2C%20step). (Misalnya, modul Futures akan mengambil funding\_rate, open\_interest, long\_short\_ratio dsb., sedangkan modul Spot tidak memiliki data tersebut[[3]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=5%29%20Derivatif%20%28Futures%E2%80%91only%29%20,harga%20padat%20likuidasi%2C%20kalau%20ada).)
* **Modul Indikator & Fitur**: Terlepas dari Spot/Futures, modul indikator (autoanalisa/indicators/mtf\_indicators.py) dan fitur teknikal (autoanalisa/features/\*) akan digunakan bersama. Fungsi-fungsi seperti penghitungan EMA, RSI, MACD, ATR, Bollinger Bands, struktur pasar (HH, HL, LH, LL), level S/R, dll., bersifat generic dan bekerja dengan data OHLCV tertutup (closed bars)[[4]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=4,autoanalisa%2Findicators%2Fmtf_indicators.py)[[5]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=Buat%20paket%20%60autoanalisa%2Ffeatures%2F%60%3A%20,%28opsional%2C%20aktifkan%20via%20flag). Dengan demikian, analisis indikator dan struktur pasar dapat digunakan baik untuk Spot maupun Futures tanpa duplikasi kode.
* **Modul Payload Builder**: Modul autoanalisa/payload/builder.py bertugas menggabungkan data dari datasource dan indikator menjadi satu **payload JSON** terstruktur sesuai skema v1[[6]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=def%20build_payload,build%28%29)[[7]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=Detail%20implementasi%20,Jika). Kelas PayloadBuilder akan memanggil modul data **spot** atau **futures** berdasarkan parameter market yang diberikan, lalu menyusun payload dengan format uniform. Hal ini menjamin bahwa meskipun data Futures memiliki kolom ekstra (funding, OI, dll.), payload akhir tetap mengikuti skema tunggal yang mencakup komponen *MTF*, *account*, *derivatives*, *orderbook*, *levels*, *market\_structure*, dll., sebagaimana didefinisikan dalam schemas/payload\_v1.py[[8]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=5%29%20Skema%20,aware). **Catatan**: Semua angka dibulatkan sesuai precision pasar (price/qty) menggunakan helper round\_to()[[2]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=,round_to%28x%2C%20step), timezone diseragamkan ke Asia/Jakarta (WIB) atau Asia/Makassar (WITA) sesuai konfigurasi[[9]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=2%29%20Arsitektur%20%26%20Konvensi%20,round_to%28x%2C%20step), dan hanya candle **closed** yang digunakan (bar yang masih berjalan dibuang)[[10]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=1.%20Ambil%20,notes).
* **Modul Rules Engine**: Logika deteksi sinyal pullback 1–4% ditempatkan di modul baru autoanalisa/rules/pullback\_v1.py. Modul ini sepenuhnya terpisah dari builder, berisi fungsi utama generate\_signals(payload: dict) -> list[Signal][[11]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%28%E2%89%A512%20assert). Ia memproses **payload JSON** yang dihasilkan builder, menerapkan serangkaian aturan (rules) trading untuk menemukan peluang entry **LONG** maupun **SHORT** sesuai kriteria pullback. Modul ini juga memanfaatkan model Pydantic Signal (didefinisikan di schemas/signal\_v1.py) untuk memastikan output sinyal terstruktur dengan benar[[12]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,memeriksa%20sinyal%20dari%20file%20payload). Dengan pemisahan ini, **payload builder** hanya fokus menyiapkan data analisis, sedangkan **rules engine** fokus pada pengambilan keputusan sinyal berdasarkan data tersebut. Arsitektur terpisah ini memudahkan perawatan: update indikator/data source tidak mengganggu logika sinyal, dan update strategi trading tidak mempengaruhi pengumpulan data.

Ringkasnya, arsitektur terdiri atas modul **datasource** Spot/Futures (mengambil data mentah), modul **indicators/features** (mengolah data teknikal), modul **payload** (mengompilasi JSON siap analisis)[[13]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=6%29%20Builder%20Utama%20File%3A%20)[[14]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=Detail%20implementasi%20,Jika), dan modul **rules** (menghasilkan sinyal trading dari JSON). Mode Spot vs Futures ditentukan sejak awal dan mempengaruhi jalur eksekusi (misal pemanggilan API dan adanya data *derivatives* khusus futures). Dengan desain ini, sistem dapat menangani kedua jenis market secara eksplisit tanpa tumpang tindih logika, sesuai prinsip “Pisahkan mode: SPOT vs FUTURES (derivatif) → data khususnya berbeda”[[15]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=,HH%2FHL%2FLH%2FLL).

## 2. Alur Eksekusi Sinyal (Dari Input JSON ke Output Sinyal JSON)

Alur eksekusi dimulai dari input **payload JSON** yang berisi data pasar multi-timeframe, kemudian diproses oleh rules engine hingga menghasilkan output **signal JSON** yang siap digunakan. Berikut langkah-langkahnya:

1. **Pembuatan Payload (Input JSON)** – Sistem menerima simbol aset dan mode market (spot/futures) sebagai input. Modul builder kemudian mengumpulkan data untuk simbol tersebut:
2. Mengambil info exchange (precision harga & qty, fee, dll) dan kondisi akun (saldo, leverage, risk\_per\_trade, dll) sesuai mode[[16]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=,%60price_precision%60%2C%20%60qty_precision).
3. Mengambil data OHLCV untuk timeframe utama (1D, 4H, 1H, 15m, minimal 200 bar per TF)[[17]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=1.%20,ATR%3A%2014)[[18]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=3%29%20Data%20Harga%20Multi%E2%80%91Timeframe%20,dif%60%2C%20%60dea%60%2C%20%60hist). Bar terakhir dicek, jika masih berjalan akan dibuang (hanya pakai bar **closed** untuk kalkulasi indikator)[[10]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=1.%20Ambil%20,notes).
4. Menghitung **indikator teknikal** per timeframe (EMA, RSI, MACD, Stoch RSI, Bollinger Bands, ATR, volume averages) dan **statistik volume**[[18]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=3%29%20Data%20Harga%20Multi%E2%80%91Timeframe%20,dif%60%2C%20%60dea%60%2C%20%60hist)[[19]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=,%60vol_last%60%2C%20%60vol_ma5%60%2C%20%60vol_ma10).
5. Menentukan **struktur pasar** (trend up/down/side, level HH, HL, LH, LL terkini) dan level-level **support/resistance** penting[[20]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=4,tf%60%2C%60dir%60%2C%60top%60%2C%60bottom).
6. Jika mode **Futures**, mengambil data **derivatif** tambahan: funding rate, next funding time, open interest, long/short ratio, mark price, index price, dll[[3]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=5%29%20Derivatif%20%28Futures%E2%80%91only%29%20,harga%20padat%20likuidasi%2C%20kalau%20ada). (Untuk Spot, bagian ini diabaikan atau fields diisi null karena tidak berlaku.)
7. Mengambil snapshot **orderbook depth** (top 5 bid/ask) untuk menghitung spread dan order book imbalance[[21]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=6%29%20Orderbook%20%26%20Eksekusi%20,taker_buy_ratio_1m%2F5m).
8. Menyusun semua komponen ke dalam struktur **PayloadV1** (Pydantic) dan menghasilkan output JSON (dengan .model\_dump())[[22]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=5.%20Susun%20blok%20,model_dump%28by_alias%3DTrue%2C%20exclude_none%3DTrue). Contoh payload minimum meliputi kunci: symbol, exchange, market, timezone, account (saldo, fee, risk, leverage jika ada), open\_position (jika ada posisi terbuka di futures), tf (indikator per timeframe), market\_structure, levels, derivatives (hanya futures), orderbook, dll[[23]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=D,1000.0)[[24]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=,19T11%3A59%3A00%2B07%3A00).

**Catatan:** Payload JSON menggantikan kebutuhan screenshot chart, sehingga datanya lengkap untuk analisis GPT[[25]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=Agar%20GPT%20bisa%20menganalisis%20,untuk%20menghasilkan%20rencana%20entry%E2%80%93TP%E2%80%93SL). Contoh struktur payload dapat dilihat pada dokumen “Checklist Data…”[[23]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=D,1000.0)[[24]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=,19T11%3A59%3A00%2B07%3A00).

1. **Pemanggilan Rules Engine** – Payload JSON yang dihasilkan kemudian diberikan ke fungsi generate\_signals(payload) dalam modul rules **pullback\_v1**[[26]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%28%E2%89%A512%20assert). Fungsi ini memeriksa peluang sinyal trading berdasarkan strategi **pullback 1–4%**. Proses internalnya mencakup beberapa tahap:
2. **Gating (Filter Utama)** – Pertama, fungsi akan melakukan **filter kondisi** awal (gating) untuk memastikan hanya mencari sinyal pada situasi berprobabilitas tinggi[[27]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=2%29%20Gating%20%28Filter%20Utama%29%20,probabilitas%20tinggi). Beberapa aturan gating misalnya:
   * **Trend Regime**: memastikan arah trend utama jelas (contoh: uptrend jika harga 1H di atas EMA50 & EMA200, downtrend sebaliknya)[[28]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=1.%20%2A%2ARegime%20Tren%2A%2A%20,%60is_trend_down%60%20%3D%20kebalikan).
   * **Likuiditas & Biaya**: spread bid-ask harus kecil (misal rasio spread < 0.05% dari harga) dan funding rate tidak ekstrem (|funding\_rate| ≤ 0.02% untuk futures)[[29]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%E2%86%92%20hindari%20short%20agresif). (Pada Spot, funding\_rate tidak ada, sehingga kondisi ini diabaikan atau dianggap lolos secara default.)
   * **Volatilitas Wajar**: ATR% 1H berada dalam rentang sehat (≥1% dan ≤8%)[[30]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=3.%20%2A%2AVolatilitas%20masuk%20akal%2A%2A%20,%E2%86%92%20hindari%20short%20agresif) – artinya tidak terlalu flat tapi juga tidak terlalu liar.
   * **Korelasi/Makro**: mempertimbangkan bias BTC/makro jika tersedia – misal, jika btc\_bias **bearish** maka hindari sinyal long agresif; jika **bullish** hindari short agresif[[31]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%E2%86%92%20hindari%20short%20agresif). (Integrasi detail bias makro dibahas di Seksi 4 di bawah.)

* Jika **salah satu** kondisi gating gagal (False), maka generate\_signals akan langsung return **no-trade** (list kosong)[[32]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=4.%20%2A%2AKorelasi%20BTC%20%28opsional%29%2A%2A%20,%E2%86%92%20hindari%20short%20agresif), artinya **tidak ada sinyal** yang diambil pada keadaan pasar tersebut. Gating ini mencegah sistem mencoba entry pada situasi berisiko rendah sukses.

1. **Build Context** – Jika lolos gating, rules engine membangun konteks analisis dari payload. Data multi-timeframe dan indikator dipersiapkan dalam struktur yang mudah diakses (misal disimpan di dict ctx). Ini mencakup nilai EMA, level-level kunci, ATR, status trend & struktur pasar tiap timeframe, dsb., yang akan digunakan oleh masing-masing rule setup. Contoh perhitungan helper: ATR% 1H dan rasio spread dihitung sekali di awal untuk digunakan di beberapa rule[[33]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=Helper%20numerik%20yang%20harus%20tersedia%3A,05%20%2F%20100).
2. **Cek Setiap Setup** – Rules engine kemudian mencoba **setiap pola/setup** pullback yang didefinisikan, baik untuk LONG maupun SHORT. Sesuai dokumen spesifikasi, minimal ada **6 setup**: **L1, L2, L3** (untuk skenario LONG) dan **S1, S2, S3** (untuk SHORT)[[34]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=S2%20%E2%80%93%20Pop%20ke%20Supply,%60%5B1.7%2C%203.0%2C%204.0)[[35]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=S3%20%E2%80%93%20False%20Break%20Round,untuk%20long%20%28variasi%20kebalikan). Tiap setup diimplementasikan sebagai fungsi (misal setup\_L1(ctx)) yang akan memeriksa kondisi **Kondisi** dan **Trigger** spesifik:
   * Contoh **L1 (Long)**: Pullback ke EMA20 1H lalu **reclaim** EMA50 15m. Kondisi: trend up (is\_trend\_up True) & harga 15m turun ke area EMA20 1H ±0.2×ATR(1H)[[36]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=L1%20%E2%80%93%20Pullback%20ke%20EMA20,%28ambil%20yang). Trigger: candle 15m close di atas EMA50 15m + terbentuk HL konfirmasi[[37]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=L1%20%E2%80%93%20Pullback%20ke%20EMA20,lebih%20longgar). Jika terpenuhi, fungsi setup mengembalikan kandidat sinyal (side "long", entry\_zone di sekitar EMA20 1H, SL di bawah HL konfirmasi atau 0.8×ATR15m, TP 1.2/2.2/3.5% etc).
   * Contoh **S1 (Short)**: Breakdown & retest gagal di 15m. Kondisi: harga di bawah EMA50 15m, membentuk lower high (LH) berurutan[[38]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=S1%20%E2%80%93%20Breakdown%20%26%20Retest,support%2015m%2F1H%20terdekat). Trigger: pullback ke EMA50 15m ±0.1×ATR15m lalu muncul candle rejection (bearish engulfing/pin bar) + konfirmasi MACD 5m melemah[[38]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=S1%20%E2%80%93%20Breakdown%20%26%20Retest,support%2015m%2F1H%20terdekat). Hasilnya kandidat sinyal short (entry di sekitar EMA50 15m, SL di atas LH atau +0.8×ATR15m, TP 1.2/2.2/3.5%).
   * (Dan seterusnya untuk setup L2, L3, S2, S3 sesuai aturan detail di dokumen rules.) Setiap setup yang terpenuhi akan menghasilkan objek **Signal** awal (kandidat) berisi parameter entry, SL, TP, dsb.
3. **Skoring & Seleksi Sinyal** – Semua kandidat sinyal yang berhasil dideteksi dari langkah sebelumnya kemudian diberi **skor** dan dievaluasi. Skoring didasarkan pada jumlah **konfluensi** yang mendukung sinyal, dengan rentang 0–100. Contoh komponen skor: lolos gating +30, terdapat multi-konfluensi (EMA, BB, S/R) +15, struktur trend mendukung (misal ada HH/HL untuk long) +15, orderflow mendukung +10, data derivatif sehat +10, konteks BTC/makro selaras +10, dengan penalti jika melawan trend utama (−15) atau dekat event volatil (−10)[[39]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=6,60). Total skor agregat menentukan kelayakan sinyal.
   * Setelah skoring, jika terdapat lebih dari satu sinyal **kandidat**, dilakukan **resolve conflict**: umumnya pilih sinyal dengan skor tertinggi. Jika ada dua sinyal berlawanan arah dengan selisih skor kecil (<5 poin), sistem dapat memutuskan **no-trade** untuk menghindari ambiguitas[[40]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%E2%88%9215%3B%20proximity%20ke%20event%2Fhigh%E2%80%91vol%20%E2%88%9210).
   * Hasil akhirnya, jika terdapat sinyal terbaik dengan score >= 60 (ambang minimal) maka fungsi generate\_signals mengembalikan list berisi sinyal tersebut (maksimum 1 sinyal)[[41]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=6,60). Jika tidak ada yang layak, fungsi mengembalikan list kosong[[42]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=14%29%20Acceptance%20Criteria%201.%20,invalid_level).
4. **Output Sinyal JSON** – Sinyal trading yang dihasilkan diformat sebagai JSON sesuai skema SignalV1. Output ini mencakup informasi lengkap untuk eksekusi atau display di frontend tanpa perlu kalkulasi tambahan. Contoh struktur output signal JSON (untuk 1 sinyal) antara lain[[43]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%5B0.882%2C%200.887)[[44]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,1H):
5. symbol: pasangan koin, market: "futures" / "spot", side: "long" atau "short".
6. setup: kode/nama setup yang terpicu (misal "L1\_ema20\_reclaim\_1h" atau sejenisnya).
7. score: skor konfluensi (0–100).
8. entry\_zone: range harga entry yang disarankan (misal [0.882, 0.887] untuk short S2)[[45]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%22%2B4.0).
9. invalid\_level: level invalidasi setup (batas dimana skenario gagal; seringkali sama dengan SL awal).
10. sl: stop-loss yang diusulkan (harga atau level invalidasi utama).
11. tp: target profit dalam persen (list string, misal ["+1.7%", "+3.0%", "+4.0%"])[[46]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,fixed_risk) beserta tp\_price kalkulasi harga absolut untuk TP tersebut sesuai precision[[44]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,1H).
12. risk\_per\_trade: porsi risiko yang dipakai (diambil dari account.risk\_per\_trade, bisa disesuaikan jika sinyal counter-trend, dll).
13. position\_sizing: metode sizing (misal "fixed\_risk" dengan perhitungan qty = risk% \* balance / range SL)[[47]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,fee_taker).
14. notes: daftar catatan kontekstual, misal alasan sinyal (contoh: ["cluster MA 1H reject", "MACD 5m roll-down"])[[48]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=%22tp%22%3A%20%5B%22%2B1.7,%7D).
15. timeframe\_confirmations: info timeframe utama konfirmasi sinyal (misal "confirm\_tf": "15m", "support\_tf": "1H" menunjukkan sinyal dikonfirmasi di 15m dengan dukungan struktur di 1H)[[49]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,1H).

Seluruh output sinyal disusun dalam bentuk JSON yang **kompatibel dengan frontend Next.js** yang sudah ada. Frontend dapat membaca field-field ini untuk ditampilkan kepada pengguna (misal entry, SL, TP, dan catatan strategi). **Tidak ada eksekusi otomatis** yang dilakukan pada tahap ini – sistem hanya memberikan rencana entry/exit dalam format terstruktur. Integrasi ke modul eksekutor trading (misalnya pembuatan order OCO dari sinyal) dapat dilakukan terpisah jika diperlukan, namun sesuai permintaan, modul AutoAnalisa ini berhenti pada output rekomendasi JSON saja (manusia/komponen lain yang mengeksekusi)[[50]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=8,fee_taker). Hal ini memastikan kontrol penuh di tangan pengguna dan mencegah aksi trade tanpa konfirmasi.

## 3. Rencana Integrasi rules\_pullback.py ke Analisis SPOT

Meskipun **rules pullback 1–4%** awalnya dirancang untuk market **Futures**, sistem perlu diadaptasi agar sinyal serupa dapat dihasilkan untuk market **Spot**. Berikut rencana integrasi dan penyesuaiannya:

* **Dukungan Payload Spot** – Fungsi generate\_signals(payload) akan dibuat kompatibel dengan payload berjenis Spot. Ini berarti code dalam rules\_pullback.py harus mempertimbangkan bahwa beberapa field khusus futures mungkin **tidak tersedia** pada payload spot. Contohnya, payload Spot tidak memiliki funding\_rate, oi, atau long\_short\_ratio (karena hanya relevan di futures)[[3]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=5%29%20Derivatif%20%28Futures%E2%80%91only%29%20,harga%20padat%20likuidasi%2C%20kalau%20ada). Solusinya:
* Pastikan saat parsing payload, jika payload['market'] == "spot", logika yang menyentuh field futures dilewatkan atau diberi **nilai default**. Misal, fungsi gating funding\_ok selalu True untuk spot (atau skip cek) karena tidak ada funding fee di spot.
* Demikian pula, check spread\_ok tetap bisa digunakan untuk Spot (spread orderbook berlaku umum), tetapi check seperti funding\_ok dan batas leverage tidak diaplikasikan. Kita bisa memasukkan handling sederhana: if payload.market == "spot": funding\_ok = True secara implisit.
* Data **derivatives** yang kosong pada payload Spot sebaiknya tidak mengganggu perhitungan skor. Skor komponen “derivatif sehat” (+10) bisa dianggap terpenuhi netral untuk spot, atau tidak dihitung sama sekali. (Implementasi: jika field tidak ada, jangan tambah skor atau penalti dari faktor tersebut.)
* **Parameter Risiko & Leverage Rendah** – Pada trading Spot (terutama jika tanpa margin atau margin rendah), pendekatan manajemen risiko sedikit berbeda:
* **Leverage**: Biasanya account.leverage tidak digunakan untuk spot (anggap 1×). Jika user menggunakan margin trading di spot dengan leverage rendah (misal 3×), kita bisa menambahkan field opsional leverage di payload spot atau cukup menetapkan bahwa sizing akan mengabaikan faktor leverage (karena di spot, posisi real tidak leveraged). Untuk konsistensi perhitungan ukuran posisi (position\_size), jika leverage tidak ada, sistem menganggap leverage = 1.
* **Risk per trade**: Nilai risk\_per\_trade dari account bisa digunakan sama (mis. 0.5–1% modal per trade), namun perlu diingat tanpa leverage risk tersebut adalah actual exposure. Jika di futures user pakai 1% risiko dengan 10× leverage, di spot mungkin tetap 1% tanpa leverage sudah lebih konservatif. Karena itu, tidak perlu mengubah risk\_per\_trade default, cukup pastikan sizing formula memperhitungkan leverage=1 sehingga hasil position size otomatis lebih kecil dibanding futures leveraged.
* **Stop-Loss Lebih Longgar**: Dikatakan *“versi leverage rendah & SL longgar untuk SPOT”*, artinya kita mengizinkan rentang SL yang lebih lebar pada spot karena tidak ada risiko likuidasi. Implementasinya, beberapa parameter aturan dapat disesuaikan:
  + Pada aturan L1, L2, S1, dll yang menggunakan faktor ATR untuk SL, kita bisa memilih batas **lebih longgar** jika ada dua opsi. Misal di L1: SL di bawah HL konfirmasi **atau** 0.8×ATR(15m), “ambil yang lebih longgar”[[51]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=pada%205%E2%80%9315m.%20,resist_15m%2C%20resist_1h%2C%20prev_high). Aturan ini sudah ada di spesifikasi, dan dapat digunakan apa adanya. Namun jika diperlukan, bisa disiapkan konfigurasi berbeda untuk spot vs futures. Contoh: di config rules\_pullback.yaml, ATR multipliers untuk SL bisa diatur sedikit lebih besar untuk spot (misal 1.0×ATR untuk spot vs 0.8×ATR untuk futures) untuk memberi ruang gerak harga. Kita dapat menambah conditional di code atau profile config terpisah.
  + Demikian pula, **trailing stop** dan breakeven rules masih berlaku, namun pada spot trader mungkin lebih fleksibel menahannya. Kita tetap output rencana trailing (misal “pindah SL ke BE di TP1”) agar disiplin tetap terjaga, tapi user bisa memilih.
* **Sinyal SHORT pada Spot** – Perlu diputuskan apakah akan mengaktifkan sinyal **short** di mode spot. Karena trading spot normal tidak bisa short tanpa margin, ada dua skenario:
* **User tidak menggunakan margin (hanya long spot)**: Dalam hal ini, sistem sebaiknya **menyaring** sinyal short saat mode spot. Implementasi sederhana: di akhir generate\_signals, jika market=="spot", buang semua kandidat side=="short". Dengan demikian output sinyal spot hanya yang long. (Bisa juga di gating: skip semua setup S1-S3 bila market spot.)
* **User menggunakan margin (short terbatas)**: Jika user memiliki fitur margin trading, sinyal short dapat diizinkan tetapi dengan **leverage rendah**. Sistem bisa tetap menghasilkan short setups pada spot, namun risk management otomatis akan cenderung lebih konservatif (risk per trade sama tapi tanpa leverage besar). Kita juga bisa tambahkan catatan pada Signal.notes untuk short spot, misal "margin trade".

Keputusan ini bisa dibuat configurable. Untuk blueprint ini, kita asumsikan user ingin **mengaktifkan juga sinyal short di Spot dengan leverage rendah**, jadi implementasi default: **tidak membedakan long/short dalam logika**, kecuali memastikan user paham bahwa short di spot butuh margin. Sebagai mitigasi, bisa tambahkan flag config allow\_short\_spot: bool jika diinginkan.

* **Integrasi ke Pipeline** – Pengintegrasian modul rules ke alur Spot berarti, saat builder selesai membuat payload dengan market="spot", kita memanggil generate\_signals dengan payload tersebut sama seperti pada futures. Kita pastikan **builder CLI/flow** mendukung kedua mode. Misal, jika sebelumnya ada tool sim\_pullback.py untuk uji sinyal dari file payload[[52]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=12%29%20CLI%20Debug%20,pullback_v1%20import%20generate_signals), kita bisa gunakan file payload spot sebagai input untuk memastikan modul rules bekerja. Selain itu, apabila AutoAnalisa dipakai secara streamlit atau bot, perlu menambahkan opsi mode spot/futures. Contohnya, jika sebelumnya bot selalu memanggil payload builder dengan market=futures dan lalu rules, sekarang logika di bot diubah: pengguna dapat memilih mode, atau sistem otomatis tahu dari simbol (bila simbol mengindikasikan spot market).
* **Penyesuaian Skor & Validasi** – Terakhir, meski aturan pullback intrinsiknya sama, kita waspadai perbedaan dinamika spot vs futures:
* Di spot mungkin gerakan harga lebih lambat (tanpa leverage efek), namun pattern teknikal berlaku sama. Syarat ATR% dan spread% sudah memastikan token terlalu sepi difilter.
* Komponen skor "Derivatif sehat" tidak relevan di spot – ini dapat dinetralkan agar tidak mengurangi total skor potensial untuk spot. Sinyal spot harus bisa mencapai skor ≥60 juga, jadi mungkin bobot skor disesuaikan atau faktor derivatif dianggap selalu netral.
* Komponen "Konteks BTC selaras" tetap penting untuk spot, bahkan mungkin lebih (karena spot sering dipengaruhi gerakan BTC secara langsung). Jadi bagian ini tetap digunakan (lihat Seksi 4 untuk peningkatan konteks makro/BTC).

Dengan integrasi di atas, modul rules\_pullback.py akan menjadi *universal* untuk kedua market. **Summary**: Saat generate\_signals dipanggil dengan payload spot, ia akan melewati gating finansial yang tidak berlaku, tetap melakukan pengecekan teknikal (trend, ATR, struktur) yang berlaku umum, dan dapat memunculkan sinyal long/short dengan parameter entry/SL/TP disesuaikan. Risiko per trade dan sizing otomatis menyesuaikan leverage=1 (kecuali ditentukan lain). Semua ini memastikan pengguna mendapatkan sinyal **pullback 1–4%** di Spot yang serupa kualitasnya dengan di Futures, hanya lebih konservatif dalam risk management sesuai karakteristik spot (tidak overleveraged, SL lebih longgar).

## 4. Validasi Tambahan Berdasarkan Time-Window Makro (WITA)

Untuk meningkatkan akurasi sinyal, sistem akan mengintegrasikan **data makro time-window** (zona waktu WITA) sebagai filter tambahan. Artinya, algoritma akan mempertimbangkan **sesi waktu dan bias makroekonomi** (mis. jam “Hijau” vs “Merah”) sebelum mengonfirmasi sinyal. Hal ini didasarkan pada temuan di *Laporan Makro BTC*, yang memetakan jam-jam tertentu dengan bias pergerakan BTC[[53]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=Ringkasan%20%20ini%20%20memetakan,RRP%29%2C%20dan%20kebijakan%20PBOC)[[54]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=5,sering%20hijau%3B%2022%3A00%E2%80%9301%3A00%20rentan%20fade). Rencana implementasinya:

* **Klasifikasi Sesi Waktu** – Pertama, kita tetapkan kategori waktu harian dalam zona waktu WITA (UTC+8) misalnya:
* **Hijau** – Sesi cenderung *bullish* (risk-on), biasanya diidentifikasi pada jam-jam dimana likuiditas mendukung kenaikan BTC. Dari laporan, contoh pola umum: **01:00–08:00 WITA sering Hijau** (pasar US tutup, likuiditas Asia masuk)[[54]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=5,sering%20hijau%3B%2022%3A00%E2%80%9301%3A00%20rentan%20fade) dan **20:30–23:00 WITA Hijau** (setelah pembukaan market US dengan data positif)[[55]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=08%3A00%E2%80%9312%3A30%20%E2%86%92%20Woles%20Hijau%2012%3A30%E2%80%9317%3A00,%E2%86%92%20Hijau%2005%3A00%E2%80%9308%3A00%20%E2%86%92%20Hijau)[[56]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=20%3A30%E2%80%9323%3A00%20%E2%86%92%20Hijau%2023%3A00%E2%80%9303%3A00%20%E2%86%92,03%3A00%E2%80%9308%3A00%20%E2%86%92%20Hijau).
* **Merah** – Sesi *bearish* atau risk-off, biasanya di jam tertentu mis. **22:00–01:00 WITA rentan Merah/fade** (akhir sesi US, potensi take-profit)[[54]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=5,sering%20hijau%3B%2022%3A00%E2%80%9301%3A00%20rentan%20fade), atau pagi Asia saat berita negatif (contoh 08:00–12:30 kadang Merah bila ada pump sebelumnya)[[57]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=3%20September%202025).
* **Woles** – Sesi lamban/sideways (volatilitas rendah) baik bias hijau maupun merah (*woles* = santai). Laporan memberi label “Woles Hijau” atau “Woles Merah” untuk periode yang pergerakannya tidak terlalu agresif meski ada bias arah[[53]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=Ringkasan%20%20ini%20%20memetakan,RRP%29%2C%20dan%20kebijakan%20PBOC).

Kategori ini bisa kita representasikan secara sederhana, misal dengan variabel session\_bias yang bisa bernilai: "bullish", "bearish", atau "neutral" (untuk woles/netral). Penentuan ini dapat dilakukan dengan aturan statis (hardcoded window) atau lebih dinamis menggunakan input kalender makro.

* **Penentuan Bias Saat Ini** – Sistem perlu mengetahui bias sesi **saat ini** atau mendatang sebelum mengambil sinyal:
* Cara sederhana: berdasarkan jam lokal (WITA) sekarang. Misal, bot mengonversi now() ke Asia/Makassar, lalu cek intervalnya termasuk kategori mana. Tabel window harian bisa disusun (mungkin dari template pola umum, atau bahkan membaca file konfigurasi macro\_sessions.yaml yang berisi jadwal hijau/merah per sesi).
* Cara lanjutan: membaca input laporan makro harian. Dokumen "Laporan Makro–Korelasi BTC" menyebut sumber seperti *Catatan ekonomi.txt (time-window WITA + katalis makro)*[[58]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=Laporan%20Makro%E2%80%93Korelasi%20BTC%20,workdir). Jika sumber ini tersedia dalam bentuk terstruktur (JSON/YAML), builder bisa mengintegrasikannya. Bahkan di **Bonus** payload builder ada opsi --news-json untuk menyisipkan event makro. Kita dapat memanfaatkan opsi tersebut: misal membuat file JSON berisi jadwal sesi hijau/merah, lalu builder menaruh ringkasan bias sebagai payload.macro\_bias atau menandai tiap timeframe dengan sesi (opsional).
* Sebagai langkah awal (MVP), boleh dimulai dengan aturan statis yang disimpulkan dari data historis:
  + 01:00–08:00 WITA = bias **bullish** (Hijau)[[54]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=5,sering%20hijau%3B%2022%3A00%E2%80%9301%3A00%20rentan%20fade),
  + 08:00–12:00 WITA = cenderung **neutral/bearish** tergantung konteks (bisa lihat data hari itu, tapi jika tidak ada input, asumsikan netral atau ikut trend teknikal saja),
  + 12:00–17:00 WITA = sering **bullish** (jam Eropa/PBOC injection, contoh 12:30–17:00 Hijau di 3 Sep)[[57]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=3%20September%202025),
  + 17:00–22:00 WITA = waspada **bearish** (sesi sore-malam, menjelang data AS, e.g. 17:00–21:45 Woles Merah menanti PMI)[[55]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=08%3A00%E2%80%9312%3A30%20%E2%86%92%20Woles%20Hijau%2012%3A30%E2%80%9317%3A00,%E2%86%92%20Hijau%2005%3A00%E2%80%9308%3A00%20%E2%86%92%20Hijau),
  + 22:00–01:00 WITA = **bearish** (risk-off fade)[[54]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=5,sering%20hijau%3B%2022%3A00%E2%80%9301%3A00%20rentan%20fade),
  + Tentu ini disesuaikan bila ada event besar (misal FOMC, NFP) – namun event besar bisa dianggap dalam kategori “Merah” karena volatilitas tak menentu.
* **Integrasi ke Gating/Scoring** – Setelah menentukan bias makro waktu, kita integrasikan sebagai filter tambahan:
* **Filter Waktu (Gating Time-Window)**: Jika saat ini masuk jendela **Merah** (bearish) dan sinyal yang muncul bertipe **LONG**, sistem sebaiknya lebih ketat. Bisa berupa:
  + **Langsung No-Trade** untuk sinyal long di window Merah kecuali ada indikasi khusus (misal divergensi ekstrim). Implementasi: di fungsi gating, tambahkan cek if session\_bias == "bearish" and candidate\_side=="long": skip (atau penalti besar).
  + Atau kurangi skor: misal penalti −10 jika long di jam risk-off. Demikian pula, hindari short agresif di jam Hijau (bullish) – short di jam Hijau mendapat penalti atau gagal gating.
  + Ini sejalan dengan catatan: *“Jika btc\_bias='bear' → hindari long agresif; btc\_bias='bull' → hindari short agresif”*[[31]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%E2%86%92%20hindari%20short%20agresif). Kita bisa menganggap session\_bias sebagai komponen penentu btc\_bias secara kasar (Hijau = bull, Merah = bear).
* **Penyesuaian Skor**: Selain gating, komponen skor juga bisa dipengaruhi. Pada spesifikasi skor ada poin *“Konteks BTC selaras +10”*[[59]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%E2%88%9215%3B%20proximity%20ke%20event%2Fhigh%E2%80%91vol%20%E2%88%9210) – konteks di sini termasuk bias makro. Jadi jika sinyal searah dengan bias sesi (misal sinyal long pada jam Hijau) maka dapat +10 tambahan. Sebaliknya, kalau melawan, sudah difilter atau minimal tak dapat bonus apapun. Juga, disebut penalti *proximity ke event/high-vol −10*[[59]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%E2%88%9215%3B%20proximity%20ke%20event%2Fhigh%E2%80%91vol%20%E2%88%9210), artinya jika masuk waktu event besar (volatil), sinyal kualitasnya diragukan. Implementasi: jika sebentar lagi ada rilis data penting (bisa diketahui dari kalender event yang disisipkan via payload opsional), kurangi skor atau tolak sinyal untuk menghindari false signal saat news.
* **Penggunaan btc\_bias**: Payload memiliki field opsional btc\_bias dan dxy\_bias[[60]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=7%29%20Kalender%2FNews%20Ringkas%20%28Opsional%29%20,%60dxy_bias). Jika laporan makro sudah meringkas bias BTC (bull/bear) secara harian, field ini bisa langsung digunakan di gating (sebagai sudah di-cover di poin gating #4 sebelumnya). Namun, btc\_bias biasanya mencerminkan trend makro umum (beberapa hari ke depan) ketimbang jam spesifik. Jadi, kita bisa mengombinasikan: misal, btc\_bias harian = bull, tapi kita juga periksa window jam. Keduanya harus selaras untuk trade agresif.
* **Implementasi Teknis**:
* Tambah modul util waktu (jika belum ada) di autoanalisa/utils/time.py untuk konversi now ke WITA dan cek range window. Misal fungsi current\_session\_bias(now\_wita) -> str yang return "bullish"/"bearish"/"neutral". Fungsi ini pakai tabel jadwal (bisa hardcode atau baca config).
* Saat generate\_signals, setelah gating teknikal, panggil fungsi ini. Jika return "bearish" sementara sinyal long, langsung return no-trade (atau continue ke next setup tanpa menambah kandidat). Begitu juga sebaliknya.
* Atau masukkan di gating sebagai kondisi terakhir:
* bias = current\_session\_bias(now\_wita)  
  if bias == "bearish" and is\_trend\_up:   
   # sesi risk-off, hindari entry long baru  
   return []  
  if bias == "bullish" and is\_trend\_down:  
   # sesi risk-on, hindari entry short  
   return []
* Logika di atas mencegah membuka posisi melawan bias makro jam tersebut, kecuali trend technical juga sejalan (kalau trend down tapi bias bullish, mungkin conflicting – kemungkinan no trade).
* Tambahkan juga info sesi di output notes sinyal untuk transparansi, misal: "notes": ["Waktu WITA: 22:30 (bias Merah)"] supaya user tahu konteks.
* **Validasi**: Uji coba filter ini terhadap data historis (backtest manual) sangat dianjurkan. Pastikan dengan filter ini, sinyal-sinyal buruk di jam "Merah" berkurang, sedangkan sinyal di jam "Hijau" tetap muncul. Sesuaikan threshold jika terlalu banyak menyaring. Jika jadwal makro berubah (misal ada hari di mana 08:00–12:00 jadi bullish karena news), perlu mekanisme update (bisa manual update config atau future improvement membaca data dari API kalender ekonomi).

Dengan langkah di atas, integrasi time-window makro WITA akan menambah lapisan konfirmasi ekstra. Sinyal hanya akan muncul ketika **kondisi teknikal dan makro mendukung bersama-sama**, meningkatkan probabilitas win-rate sesuai target (65–75% win rate, PF >2 seperti disebut di dokumen)[[61]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,dan%20Profit%20Factor%20%3E2).

## 5. Langkah Patch & Cakupan Pengujian

Setelah perancangan di atas, berikut rencana implementasi (*patch plan*) secara bertahap disertai rencana pengujian untuk memastikan setiap komponen berfungsi benar:

**Langkah-langkah Implementasi:**

1. **Persiapan Branch & Lingkungan** – Buat branch git baru, misal feat/autoanalisa-spot-pullback, untuk mengisolasi perubahan. Pastikan semua dependensi yang dibutuhkan sudah ada di requirements.txt (misal pydantic>=2.7, pytz, dll sudah ditambahkan sesuai instruksi sebelumnya[[62]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=1%29%20Dependensi%20Tambahkan%20ke%20,%28Boris%29%20untuk)). Setup kerangka direktori modul jika belum (pastikan paket autoanalisa/ sudah terstruktur dengan sub-package datasource/, indicators/, features/, payload/, rules/, schemas/, utils/, dll., dan file \_\_init\_\_.py di masing-masing[[63]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=,ts%7D.json)).
2. **Review & Refactor Modul Builder** – Pastikan implementasi PayloadBuilder sesuai spesifikasi:
3. Cek bahwa market parameter sudah memicu pemanggilan fungsi data yang benar (spot vs futures). Jika belum, implementasikan conditional: if market=="spot": use binance\_spot.py; elif "futures": use binance\_futures.py.
4. Tambahkan support untuk opsional data makro: misal, modul builder bisa mengecek keberadaan file konfigurasi sesi makro. (Bonus opsional: implement --news-json arg pada CLI payload builder untuk inject event macro, atau lebih sederhana: sisipkan pembacaan btc\_bias dari config harian jika ada.)
5. Validasi output payload: jalankan python tools/build\_payload.py <symbol> --market spot dan --market futures untuk beberapa simbol. Periksa JSON output sesuai skema: ada 4 timeframe, timezone benar, data futures lengkap (jika futures), data spot tidak mengandung field aneh. **Pastikan tidak error** meski field futures kosong pada spot (harus di-exclude\_none=True saat dump[[22]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=5.%20Susun%20blok%20,model_dump%28by_alias%3DTrue%2C%20exclude_none%3DTrue)).
6. **Implementasi Rules Pullback v1** – Buat file autoanalisa/rules/pullback\_v1.py apabila belum ada, lalu implementasikan sesuai dokumen “Instruksi Codex GPT – Rules Futures Pullback...”:
7. Definisikan fungsi generate\_signals(payload) dengan struktur: gating -> build ctx -> cek setup L/S -> scoring -> conflict resolution -> return signals[[64]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=9%29%20API%20Fungsi%20,signal_v1%20import%20Signal)[[65]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,append%28score_signal%28sig%2C%20ctx).
8. Implementasi detail tiap setup (L1..S3) sesuai rule. Gunakan data dari payload (via ctx) untuk cek kondisi trigger (contoh: EMA20 1H didapat dari payload['tf']['1H']['ema']['20'], dsb).
9. Masukkan toleransi faktor ATR, level reclaim, dll sesuai panduan dokumen[[37]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=L1%20%E2%80%93%20Pullback%20ke%20EMA20,lebih%20longgar)[[38]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=S1%20%E2%80%93%20Breakdown%20%26%20Retest,support%2015m%2F1H%20terdekat).
10. Pastikan struktur Signal (Pydantic) dibuat di schemas/signal\_v1.py sesuai format[[26]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%28%E2%89%A512%20assert)[[43]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%5B0.882%2C%200.887), termasuk alias nama field jika diperlukan. Uji serialisasi .model\_dump() menghasilkan JSON field yang diinginkan (snake\_case atau camelCase sesuai frontend, kemungkinan snake\_case sesuai contoh).
11. *Note*: Jika modul ini sudah pernah diimplementasi (misal via Codex GPT), fokuskan pada modifikasi berikutnya untuk Spot dan makro.
12. **Adaptasi untuk Spot** – Modifikasi pullback\_v1.py untuk mendukung Spot:
13. Di bagian **gating**, tambahkan handling:
    * Jika payload['market']=="spot", set funding\_ok = True tanpa cek (atau skip sama sekali).
    * Kondisi spread\_ok tetap dihitung (spread spot juga penting).
    * ATR% filter bisa sama.
    * Trend regime, struktur: sama.
14. Sesuaikan logic setups bila perlu: kebanyakan aturan (L1–S3) tetap berlaku karena murni teknikal (harga dan indikator). Pastikan tidak ada penggunaan variabel khusus futures (contoh: mungkin S2 menyebut *MA cluster 1H di zona X* bisa dipakai di spot juga karena MA ada).
15. Integrasikan parameter risk/leverage: untuk kalkulasi position size atau output risk\_per\_trade, jika leverage tak ada (spot) maka leverage diasumsikan 1 dalam formula qty = (balance \* risk) / (|entry - SL|). Bisa membuat helper di utils/num.py seperti yang disarankan, dan gunakan secara konsisten.
16. Berikan kondisi penyesuaian short spot: jika kita memutuskan menonaktifkan short di spot (sesuai diskusi sebelumnya), lakukan filter di sini (misal hapus kandidat if payload['market']=="spot" and sig.side=="short": continue). Jika tetap mengizinkan, mungkin tambahkan note di sinyal.
17. **Integrasi Time-Window Makro** – Implementasikan util dan integrasikan:
18. Tambah fungsi waktu di utils/time.py untuk mendapatkan waktu WITA saat ini. Gunakan datetime.now(tz) dengan tz=Asia/Makassar (bisa manfaatkan pytz yang sudah ditambahkan[[66]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=%60%60%60%20pydantic%3E%3D2.7%20pytz%20python,binance%3D%3D1.0.20%2C%20pandas%2C%20numpy%2C%20ta)). Atau, gunakan timestamp terakhir di payload (misal payload['tf']['15m']['close\_time']) sebagai referensi waktu analisis.
19. Buat tabel window (sementara hardcode) dalam fungsi atau baca dari file jika disiapkan. Contoh sederhana:

* WITA\_SCHEDULE = [  
   ("01:00", "08:00", "bullish"),  
   ("22:00", "01:00", "bearish"),  
   # ... other windows ...  
  ]
* Perhatikan interval yang melewati tengah malam (22:00–01:00) perlu dipecah atau dihandle logika wrap.

1. Implement get\_session\_bias(dt\_wita) -> str yang mereturn "bullish"/"bearish"/"neutral" berdasarkan jam di dt\_wita.
2. Panggil fungsi ini di generate\_signals sebelum loop setup (atau di gating). Misal:

* bias = get\_session\_bias(now\_wita)  
  if bias == "bearish" and trend\_main == "up":  
   return [] # hindari long saat bias makro bearish  
  if bias == "bullish" and trend\_main == "down":  
   return [] # hindari short saat bias makro bullish
* (Gunakan juga info btc\_bias dari payload jika ada: bisa override bias = 'bullish'/'bearish' based on that, atau kombinasikan).

1. Selain gating, tambahkan efek ke skor: misal, di fungsi score\_signal(sig, ctx) tambahkan:

* if sig.side == "long" and bias == "bullish": score += 10  
  elif sig.side == "short" and bias == "bearish": score += 10  
  elif sig.side == "long" and bias == "bearish": score -= 10 # or simply fail earlier
* Sesuaikan besaran skor agar total tetap max 100.

1. Uji cepat: misal set waktu ke jam yang berbeda dan lihat apakah sinyal di-filter dengan benar.
2. **Testing (Unit Tests)** – Buat / perbarui test di tests/:
3. **Test Payload Builder**: Siapkan data OHLCV dummy (CSV) untuk spot dan futures (misal pair IMXUSDT) seperti yang dicontohkan[[67]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=8%29%20Unit%20Test%20Minimal%20,vol_last). Monkeypatch fungsi API call agar pakai data dummy. Lakukan unit test build\_payload(symbol, "futures") dan build\_payload(symbol, "spot"). **Assert** bahwa:
   * Output JSON memiliki struktur lengkap. Untuk futures, field derivatives terisi (funding\_rate dsb mungkin dummy value); untuk spot, field derivatives mungkin absen atau null tapi tidak menyebabkan error.
   * Setiap timeframe memuat indikator utama (EMA/RSI/MACD dll)[[68]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=,150%20%E2%86%92%20%60insufficient_history%3D%3DTrue).
   * close\_time format ISO8601 timezone-aware[[69]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=,150%20%E2%86%92%20%60insufficient_history%3D%3DTrue).
   * Jika data history kurang (df<150) maka payload insufficient\_history=True ditandai[[69]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=,150%20%E2%86%92%20%60insufficient_history%3D%3DTrue).
4. **Test Gating**: Buat payload mock yang melanggar satu per satu aturan gating:
   * Misal **funding\_rate** di atas threshold (0.0005 > 0.0002) -> generate\_signals(payload) harus return [] (no signal)[[70]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%E2%86%92%20hindari%20short%20agresif).
   * Spread terlalu besar (misal 0.1% >0.05%) -> no signal.
   * ATR% terlalu kecil atau besar (misal 0.5% atau 10%) -> no signal.
   * Macro bias conflict: set payload.bias = bear + buat scenario long -> no signal.
5. **Test Setup Logic**: Buat skenario dummy di mana kondisi salah satu setup terpenuhi, periksa apakah sinyal dihasilkan sesuai harapan:
   * Contoh **L1**: Payload diset sedemikian rupa: trend\_main up, harga 1H sekitar EMA20, harga 15m turun menembus EMA20 1H lalu 15m close > EMA50 (bisa simulate dengan angka). Pastikan generate\_signals mengembalikan satu sinyal side=="long", setup=="L1...", SL < entry, TP list ada dan di kisaran 1-4% relatif[[37]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=L1%20%E2%80%93%20Pullback%20ke%20EMA20,lebih%20longgar).
   * **S1**: Buat payload: trend\_main down, harga di bawah EMA50 15m, terjadi LH, dll. Pastikan sinyal short muncul.
   * Cek juga perhitungan TP% ke harga betul (bandingkan tp vs tp\_price dihitung dari entry dengan ±% yang diberikan).
6. **Test Scoring & Conflict**: Buat beberapa kandidat sinyal dalam satu payload:
   * Misal long dan short muncul bersamaan dengan skor selisih kecil. Pastikan fungsi conflict resolution memilih none (no trade) jika selisih <5[[65]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,append%28score_signal%28sig%2C%20ctx).
   * Jika selisih besar, pilih yang terbesar.
   * Uji penalti: misal input sinyal melawan trend 4H, skor final harus berkurang 15.
7. **Test Spot Specific**: Buat payload spot dengan kondisi mirip L1 atau S1:
   * Karena funding\_rate absen, pastikan code tidak error (gating lewat).
   * Cek kalau short di spot dihasilkan, apakah sesuai keputusan (misal kita tidak memfilter, maka sinyal short keluar; jika kita memfilter, pastikan list kosong untuk short scenario).
   * Periksa nilai position\_sizing/qty jika bisa dihitung: dengan leverage=1, hitung manual apakah formula qty sesuai risk% \* balance / (entry−SL).
8. **Test Macro Filter**: Ini bisa dilakukan dengan menyimulasikan waktu. Mungkin perlu injeksi dependency (misal fungsi get\_session\_bias dibuat dapat dioverride atau parameter waktu).
   * Uji ketika bias = bearish: masukkan pada payload btc\_bias="bear" atau set waktu di fungsi ke jam merah, dan buat scenario yang *would be* valid technical. Hasil harus no signal.
   * Uji bias bullish: sebaliknya, pastikan sinyal lolos.
   * Juga test bahwa note atau score adjustment terjadi (kalau kita menambahkan note "bias Merah" misalnya, periksa notes output mengandung itu).

Jalankan semua test dengan pytest -q. **Target**: semua unit test lulus (bebas error) dan semua assert terpenuhi, menandakan integrasi spot & macro tidak merusak fungsionalitas existing. Minimum harus ada ≥12 assert unik yang mencakup kasus di atas (sesuai spesifikasi)[[71]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%28%E2%89%A512%20assert)[[72]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=10%29%20Unit%20Test%20%28contoh%29%20,precision.price).

1. **Documentasi & Review**: Perbarui README atau dokumen usage bila perlu, terutama untuk menjelaskan pilihan mode Spot vs Futures dan interpretasi output. Buat contoh output JSON baru untuk spot dan futures setelah perbaikan, agar tim/front-end bisa menyesuaikan bila ada perubahan kecil struktur (idealnya struktur tetap sama).
2. **Pull Request & Merge**: Setelah internal testing memadai, buka PR ke branch main/master. Ikuti template patch seperti sebelumnya: sebutkan fitur yang ditambah (support spot, macro filter, etc), lampirkan hasil sample JSON, dan catat bahwa output sinyal tetap kompatibel dengan front-end Next.js (JSON structure unchanged, no auto execution). Minta code review khususnya untuk bagian perhitungan risk dan integrasi makro agar tidak ada bug logika. Setelah approve, merge ke main.

**Cakupan Pengujian Tambahan (Manual/Integration):**

Selain unit test, disarankan uji integrasi end-to-end manual: - Jalankan builder + rules secara keseluruhan: misal buat script kecil yang memanggil build\_payload(symbol, market) lalu langsung pass ke generate\_signals. Coba untuk beberapa simbol aktif di waktu berbeda (pagi vs malam) untuk melihat variasi output. - Perhatikan apakah sinyal yang dihasilkan masuk akal dengan kondisi market nyata (verifikasi kasar dengan chart). Misal, coba pada data BTCUSDT spot vs futures untuk periode di laporan makro (2–5 Sep 2025) dan lihat apakah sistem menghindari trade di window merah seperti diharapkan. - Pastikan juga output JSON dapat dibaca oleh komponen front-end tanpa error. Karena format mostly sama, seharusnya OK. Jika menambahkan field baru (contoh session\_bias), komunikasikan ke front-end (atau bisa kita letakkan di notes saja agar front-end tidak perlu perubahan struktur).

Dengan patch dan pengujian di atas, kita memastikan bahwa sistem AutoAnalisa yang diperbarui berfungsi sesuai tujuan: mendukung **Spot & Futures** secara eksplisit, mendeteksi sinyal **pullback 1–4% Long/Short** dengan tambahan filter **makro time-window WITA** demi kualitas sinyal, dan output tetap berupa JSON terstruktur untuk di-review/eksekusi user.

## 6. Prioritas Fitur dan Dependensi

Pada tahap implementasi, urutan pengerjaan dan prioritas fitur perlu diatur agar efisien dan menghindari konflik. Berikut prioritas dan dependensi antar komponen:

1. **Fondasi Arsitektur Terpisah** – Prioritas pertama adalah memastikan struktur proyek siap: modul baru untuk payload & rules telah dibuat, pemisahan Spot/Futures di data source berjalan. Ini fundamental karena tanpa ini fitur lainnya tidak bisa berdiri. Dependensi: konfigurasi environment (API keys jika perlu, library ta/pandas untuk indikator sudah ada), dan memastikan tidak ada error import/module path.
2. **Payload Builder & Schema** – Kedua, selesaikan implementasi **payload builder** (kalau belum). Payload harus mencakup semua data minimal[[73]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=B,%60price_precision%60%2C%20%60qty_precision)[[18]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=3%29%20Data%20Harga%20Multi%E2%80%91Timeframe%20,dif%60%2C%20%60dea%60%2C%20%60hist) yang diperlukan rules. Fitur ini prioritas tinggi sebab output-nya menjadi input rules. Pastikan schema Pydantic PayloadV1 dan SignalV1 tersedia sejak awal (bisa stub kalau belum final) untuk mencegah error serialisasi belakangan. *Dependensi*: API Binance (pastikan koneksi, atau gunakan mode offline data untuk dev/test), modul indikator (ta library) untuk hitung indikator.
3. **Rules Pullback 1–4% (Futures)** – Implementasi rules engine sesuai spesifikasi futures adalah inti logic trading. Prioritaskan mengkoding kondisi **L1...S3** seakurat mungkin sesuai dokumen[[37]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=L1%20%E2%80%93%20Pullback%20ke%20EMA20,lebih%20longgar)[[38]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=S1%20%E2%80%93%20Breakdown%20%26%20Retest,support%2015m%2F1H%20terdekat), karena ini tulang punggung sinyal. Mulai dari gating hingga scoring. *Dependensi*: membutuhkan payload terstruktur dengan field yang diharapkan (oleh karena itu step 2 harus benar). Juga membutuhkan model Signal untuk output.
4. **Testing & Tune (Futures)** – Sebelum lanjut ke spot, lakukan pengujian menyeluruh pada rules di mode Futures. Pastikan semua gating & setup berfungsi di lingkungan futures. Ini penting agar ketika menambahkan spot, kita tahu jika error berasal dari modifikasi baru atau sudah ada sebelumnya. *Dependensi*: Data historis untuk test, scenario design.
5. **Integrasi Spot** – Setelah futures mode stabil, baru mulai adaptasi ke Spot. Prioritas di sini mencakup:
6. Handling field differences (funding, OI, etc) – relatif mudah, prioritas tinggi sebab tanpa ini code bisa crash di spot mode.
7. Risk & leverage adjustments – penting untuk akurasi sizing tapi bisa dianggap fine-tuning; prioritas menengah (fungsionalitas sinyal tidak terganggu jika ini sedikit meleset, tapi sebaiknya benar).
8. Short signal toggle – prioritas menengah; default bisa membiarkan short, namun baiknya dipastikan sebelum rilis. Ini juga tergantung kebijakan user (jika tidak ingin short, bisa belakangan menambah filter).
9. *Dependensi*: Sukses implementasi futures rules, karena spot largely reuse logic. Juga perlu memastikan builder spot sudah memberikan data yang cukup (indikator dll tersedia, struktur market ada).
10. **Makro Time-Window Filter** – Integrasi konteks makro diberikan prioritas **tinggi** dari sisi nilai (karena ini fitur baru yang meningkatkan quality), namun secara pengerjaan dapat dilakukan **setelah** core sinyal selesai. Ini karena macro filter relatif terisolasi (hanya menambah kondisi di gating/scoring). Jadi urutan pengerjaan: setelah spot rules jalan, implement macro filter. *Dependensi*: modul waktu (pytz) dan mungkin data jadwal (bisa statis). Pastikan timezone WITA vs WIB sinkron dengan timestamp di payload. Juga jika ingin ambil bias dari payload (btc\_bias), pastikan payload builder menangani itu (opsional input event/news).
11. **Testing (Spot & Macro)** – Prioritas sangat tinggi untuk testing final di kombinasi fitur: spot + futures + macro. Jangan sampai filter makro baru malah mem-block semua sinyal, atau logic spot merusak futures. Regression test pada futures scenarios sangat penting di sini.
12. *Dependensi*: test-case disiapkan, kemungkinan perlu menyesuaikan expected output bila scoring berubah karena macro (misal sebelumnya dapat signal, sekarang no-trade karena bias, harus dianggap benar sesuai perubahan).
13. Juga pastikan test environment ada timezone awareness (bisa set timezone secara konsisten dalam code saat membandingkan timestamp).
14. **Dokumentasi & Deployment** – Terakhir, update dokumentasi (README, contoh JSON) dan informasikan tim frontend mengenai perubahan apapun (meski struktur JSON diusahakan sama, tapi ada tambahan interpretasi seperti notes atau bias). Prioritas dokumentasi tetap ada namun setelah kode benar. *Dependensi*: semua fitur selesai diimplementasi.

**Dependensi Eksternal**: Secara umum, fitur-fitur di atas menggunakan library yang sudah ada kecuali mungkin data makro. Jika tidak tersedia API untuk macro bias, kita mengandalkan konfigurasi manual. Tidak ada penambahan library baru besar, hanya gunakan yang sudah ditambahkan (pydantic, pytz). Akses internet (untuk Binance API) diperlukan untuk data live; jika digunakan offline (untuk test) kita pakai dummy CSV. Frontend Next.js konsumsi JSON via filesystem atau API — tidak terpengaruh kecuali format berubah (yang kita jaga konstan).

**Prioritas Summary**: - *Highest priority*: Memastikan kerangka **payload** dan **rules** lengkap & terpisah (Spot/Futures). - *High*: Implementasi rules pullback (logic L/S) akurat dan filter makro (karena ini core improvement target). - *Medium*: Adaptasi spot details (sizing, short toggle) – penting tapi bisa di-refine iteratif. - *Medium*: Extensive testing & validation – waktu harus dialokasikan cukup karena ini integrasi multi-komponen. - *Low*: Bonus/opsional seperti caching klines, integrasi news JSON (bila sempat, sesuai bonus poin di dokumen).

Dengan prioritas yang jelas, tim dapat fokus menyelesaikan pondasi dahulu sebelum fitur tambahan, memastikan tiap tahap berfungsi baik sebelum melangkah ke tahap selanjutnya. Hal ini akan menghasilkan sistem AutoAnalisa yang **stabil, modular, dan mampu menghasilkan sinyal pullback 1–4%** di berbagai kondisi (Spot/Futures, Long/Short) dengan mempertimbangkan **konteks makro**, sesuai tujuan perbaikan.

[[1]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=3.1%20%60binance_spot.py%60%20Fungsi%3A%20,tidak%2C%20gunakan%20default%20dari%20config) [[2]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=,round_to%28x%2C%20step) [[4]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=4,autoanalisa%2Findicators%2Fmtf_indicators.py) [[5]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=Buat%20paket%20%60autoanalisa%2Ffeatures%2F%60%3A%20,%28opsional%2C%20aktifkan%20via%20flag) [[6]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=def%20build_payload,build%28%29) [[7]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=Detail%20implementasi%20,Jika) [[8]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=5%29%20Skema%20,aware) [[9]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=2%29%20Arsitektur%20%26%20Konvensi%20,round_to%28x%2C%20step) [[10]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=1.%20Ambil%20,notes) [[13]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=6%29%20Builder%20Utama%20File%3A%20) [[14]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=Detail%20implementasi%20,Jika) [[22]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=5.%20Susun%20blok%20,model_dump%28by_alias%3DTrue%2C%20exclude_none%3DTrue) [[62]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=1%29%20Dependensi%20Tambahkan%20ke%20,%28Boris%29%20untuk) [[63]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=,ts%7D.json) [[66]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=%60%60%60%20pydantic%3E%3D2.7%20pytz%20python,binance%3D%3D1.0.20%2C%20pandas%2C%20numpy%2C%20ta) [[67]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=8%29%20Unit%20Test%20Minimal%20,vol_last) [[68]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=,150%20%E2%86%92%20%60insufficient_history%3D%3DTrue) [[69]](file://file-DusYCjJJ7DnDdyJ5TfH8rq#:~:text=,150%20%E2%86%92%20%60insufficient_history%3D%3DTrue) instruksi\_codex\_gpt\_payload\_auto\_analisa\_futures\_spot.md

<file://file-DusYCjJJ7DnDdyJ5TfH8rq>

[[3]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=5%29%20Derivatif%20%28Futures%E2%80%91only%29%20,harga%20padat%20likuidasi%2C%20kalau%20ada) [[15]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=,HH%2FHL%2FLH%2FLL) [[16]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=,%60price_precision%60%2C%20%60qty_precision) [[17]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=1.%20,ATR%3A%2014) [[18]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=3%29%20Data%20Harga%20Multi%E2%80%91Timeframe%20,dif%60%2C%20%60dea%60%2C%20%60hist) [[19]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=,%60vol_last%60%2C%20%60vol_ma5%60%2C%20%60vol_ma10) [[20]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=4,tf%60%2C%60dir%60%2C%60top%60%2C%60bottom) [[21]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=6%29%20Orderbook%20%26%20Eksekusi%20,taker_buy_ratio_1m%2F5m) [[23]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=D,1000.0) [[24]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=,19T11%3A59%3A00%2B07%3A00) [[25]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=Agar%20GPT%20bisa%20menganalisis%20,untuk%20menghasilkan%20rencana%20entry%E2%80%93TP%E2%80%93SL) [[60]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=7%29%20Kalender%2FNews%20Ringkas%20%28Opsional%29%20,%60dxy_bias) [[73]](file://file-H9uc67nwmFisQE11rGKfRj#:~:text=B,%60price_precision%60%2C%20%60qty_precision) checklist\_data\_untuk\_gpt\_pengganti\_screenshot\_chart\_futures\_spot.md

<file://file-H9uc67nwmFisQE11rGKfRj>

[[11]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%28%E2%89%A512%20assert) [[12]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,memeriksa%20sinyal%20dari%20file%20payload) [[26]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%28%E2%89%A512%20assert) [[27]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=2%29%20Gating%20%28Filter%20Utama%29%20,probabilitas%20tinggi) [[28]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=1.%20%2A%2ARegime%20Tren%2A%2A%20,%60is_trend_down%60%20%3D%20kebalikan) [[29]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%E2%86%92%20hindari%20short%20agresif) [[30]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=3.%20%2A%2AVolatilitas%20masuk%20akal%2A%2A%20,%E2%86%92%20hindari%20short%20agresif) [[31]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%E2%86%92%20hindari%20short%20agresif) [[32]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=4.%20%2A%2AKorelasi%20BTC%20%28opsional%29%2A%2A%20,%E2%86%92%20hindari%20short%20agresif) [[33]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=Helper%20numerik%20yang%20harus%20tersedia%3A,05%20%2F%20100) [[34]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=S2%20%E2%80%93%20Pop%20ke%20Supply,%60%5B1.7%2C%203.0%2C%204.0) [[35]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=S3%20%E2%80%93%20False%20Break%20Round,untuk%20long%20%28variasi%20kebalikan) [[36]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=L1%20%E2%80%93%20Pullback%20ke%20EMA20,%28ambil%20yang) [[37]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=L1%20%E2%80%93%20Pullback%20ke%20EMA20,lebih%20longgar) [[38]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=S1%20%E2%80%93%20Breakdown%20%26%20Retest,support%2015m%2F1H%20terdekat) [[39]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=6,60) [[40]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%E2%88%9215%3B%20proximity%20ke%20event%2Fhigh%E2%80%91vol%20%E2%88%9210) [[41]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=6,60) [[42]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=14%29%20Acceptance%20Criteria%201.%20,invalid_level) [[43]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%5B0.882%2C%200.887) [[44]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,1H) [[45]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%22%2B4.0) [[46]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,fixed_risk) [[47]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,fee_taker) [[48]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=%22tp%22%3A%20%5B%22%2B1.7,%7D) [[49]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,1H) [[50]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=8,fee_taker) [[51]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=pada%205%E2%80%9315m.%20,resist_15m%2C%20resist_1h%2C%20prev_high) [[52]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=12%29%20CLI%20Debug%20,pullback_v1%20import%20generate_signals) [[59]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%E2%88%9215%3B%20proximity%20ke%20event%2Fhigh%E2%80%91vol%20%E2%88%9210) [[61]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,dan%20Profit%20Factor%20%3E2) [[64]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=9%29%20API%20Fungsi%20,signal_v1%20import%20Signal) [[65]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,append%28score_signal%28sig%2C%20ctx) [[70]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%E2%86%92%20hindari%20short%20agresif) [[71]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=,%28%E2%89%A512%20assert) [[72]](file://file-33aeZvm4gGDSqU4Sc9rv3x#:~:text=10%29%20Unit%20Test%20%28contoh%29%20,precision.price) instruksi\_codex\_gpt\_rules\_futures\_pullback\_1\_4\_long\_short.md

<file://file-33aeZvm4gGDSqU4Sc9rv3x>

[[53]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=Ringkasan%20%20ini%20%20memetakan,RRP%29%2C%20dan%20kebijakan%20PBOC) [[54]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=5,sering%20hijau%3B%2022%3A00%E2%80%9301%3A00%20rentan%20fade) [[55]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=08%3A00%E2%80%9312%3A30%20%E2%86%92%20Woles%20Hijau%2012%3A30%E2%80%9317%3A00,%E2%86%92%20Hijau%2005%3A00%E2%80%9308%3A00%20%E2%86%92%20Hijau) [[56]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=20%3A30%E2%80%9323%3A00%20%E2%86%92%20Hijau%2023%3A00%E2%80%9303%3A00%20%E2%86%92,03%3A00%E2%80%9308%3A00%20%E2%86%92%20Hijau) [[57]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=3%20September%202025) [[58]](file://file-Fgb9gDQ9WNCAyfkqkGmWip#:~:text=Laporan%20Makro%E2%80%93Korelasi%20BTC%20,workdir) Laporan Makro–korelasi Btc (2–5 Sep 2025, Wita).pdf

<file://file-Fgb9gDQ9WNCAyfkqkGmWip>