

Berikut panduan ringkas untuk menambah CEX, DEX, dan CHAIN di struktur saat ini, beserta lokasi file dan langkahnya.

CEX

- Lokasi utama:
 - Definisi warna dasar: config/config.json (bagian CONFIG_CEX)
 - Registrasi + link builder: services/cex/registry.js
 - Orderbook + parser (default/fallback): services/cex.js
 - API keys (opsional, untuk wallet fetch): secrets.js
- Langkah tambah CEX:
 - Tambah warna (opsional): di config/config.json → CONFIG_CEX.MYEX: {
"WARNA": "#aabbcc" }
 - Registrasi via registry agar link trade/WD/DP terpusat:
 - Tambahkan di services/cex/registry.js:
 - Contoh:
 - CEX.register({ name: 'KUCOIN', color: '#0aa' }) minimal
 - Atau lengkap:
 - CEX.register({ name: 'KUCOIN', color: '#0aa',
linkBuilder: (token, pair) => ({ tradeToken: '...',
tradePair: '...', withdrawUrl: '...', depositUrl: '...' })
})
 - Setelah terdaftar, UI/telemetry akan gunakan GeturlExchanger yang sudah memanggil CEX.link.buildAll().
 - Orderbook parser (opsional/override):
 - Registry sudah mendukung: CEX.register({ name, orderbook: { urlTpl: coins => '...', parser: (resp)=>processOrderBook(resp,3) }})
 - services/cex.js otomatis merge exchangeConfig dengan yang ada di registry.
 - Wallet status (perlu API-exchange khusus):
 - Saat ini services/cex.js memakai fetchWalletStatus(cex) berbasis switch untuk BINANCE/MEXC/GATE/INDODAX.
 - Untuk CEX baru: tambahkan cabang baru di fetchWalletStatus dan masukkan ApiKey/Secret di secrets.js → CEX_SECRETS.MYEX.
- Kunci konsistensi:
 - Nama CEX uppercase (contoh: BINANCE, MEXC).
 - Warna bisa di config/config.json atau dibiarkan mengikuti CEX.register({ color }).

DEX

- Lokasi utama:
 - Builder link DEX default: dex-config.js
 - Registrasi DEX (builder/fallback): services/dex.js (API DEX.register())
 - Strategi harga (request/parse): services/dex.js (objek dexStrategies)
- Langkah tambah DEX:
 - Builder link + fallback policy:
 - Pilihan A (statis): tambahkan entry di dex-config.js
 - Pilihan B (runtime): panggil DEX.register('foo', { builder: ({ chainCode, tokenAddress, pairAddress }) => '...', allowFallback: true })
 - Catatan: DEX.register otomatis sinkron ke window.CONFIG_DEXS agar pemanggil lama tetap jalan.

- Strategi harga (opsional, jika ingin dipakai scanner):
 - Tambahkan dexStrategies.foo = { buildRequest(...){...}, parseResponse(resp,ctx){ return { amount_out, FeeSwap, dexTitle: 'FOO' } } } di services/dex.js.
- Aktifkan di chain:
 - Tambahkan string key DEX (lowercase) ke array DEXS di chain yang relevan (lihat bagian Chain di bawah).

CHAIN

- Lokasi utama:
 - Definisi chain: config/config.json (bagian CONFIG_CHAINS)
 - Helper akses: services/chain.js
- Struktur minimal chain baru di config/config.json:
 - Kode_Chain (chainId), Nama_Chain, Nama_Pendek, URL_Chain (explorer), ICON, WARNA
 - RPC, GASLIMIT (dipakai hitung fee swap), BaseFEEDEX
 - DEXS: array nama DEX (lowercase) yang didukung chain ini, mis. ["1inch","odos","kyberswap"]
 - WALLET_CEX: mapping alamat deposit/WD per CEX, mis. BINANCE, MEXC...
 - PAIRDEXS: mapping pair default, pastikan ada minimal USDT dan NON
- Contoh sangat ringkas:
 - "mychain": { "Kode_Chain": 1234, "Nama_Chain": "mychain", "URL_Chain": "https://explorer...", "WARNA": "#123456", "RPC": "https://...", "GASLIMIT": 100000, "DEXS": ["1inch","odos"], "WALLET_CEX": { "BINANCE": { "address": "0x..." } }, "PAIRDEXS": { "USDT": { "symbolPair": "USDT", "scAddressPair": "0x...", "desPair": "6" }, "NON": { "symbolPair": "NON", "scAddressPair": "0x", "desPair": "18" } } }
- Helper yang tersedia:
 - Chain.get('mychain') → info chain
 - Chain.listDex('mychain') → daftar DEX aktif di chain
 - Chain.cexWallet('mychain','BINANCE') → alamat CEX untuk chain

Catatan penting

- Kasing/konvensi:
 - CEX: uppercase (BINANCE)
 - DEX: lowercase keys ('kyberswap', '1inch', '0x', 'odos', 'okx', 'lifi')
 - CHAIN: lowercase key ('bsc', 'polygon', ...)
- API Key/Secret:
 - Letakkan di secrets.js → CEX_SECRETS (jangan commit ke repo publik).
- Loader:
 - config-loader.js memuat services/chain.js, services/cex/registry.js, dex-config.js, lalu modul lain. Tidak perlu ubah <script> di HTML.

Jika Anda mau, saya bisa:

- Tambahkan contoh nyata: pendaftaran KUCOIN di services/cex/registry.js.
- Tambahkan DEX fiktif fooswap di services/dex.js + builder-nya.
- Tambahkan chain contoh opbnb ke config/config.json.