

PROJECT MATA KULIAH AI FOR TRADING
Pengembangan Sistem Optimasi Trading Otomatis Berbasis *Hybrid Random Forest*
dan *Aggressive Pyramiding* pada Multi-Aset Teknologi



Disusun oleh:
Putu Gde Kenzie Carlen Mataram - 71230994
Tara Tirtanata - 71231056

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2025

Daftar Isi

BAB I.....	1
PENDAHULUAN.....	1
A. LATAR BELAKANG.....	1
B. RUMUSAN MASALAH.....	1
C. TUJUAN PENELITIAN.....	1
D. BATASAN MASALAH.....	2
BAB II.....	3
SPESIFIKASI TEKNIS PERANGKAT LUNAK.....	3
A. PUSTAKA INTI.....	3
B. KONFIGURASI LINGKUNGAN.....	4
BAB III.....	5
METODOLOGI RISET.....	5
A. PENGUMPULAN DATA DAN PRE-PROCESSING.....	5
B. FEATURE ENGINEERING.....	5
C. DESAIN MACRO MACHINE LEARNING ENGINE.....	5
D. LOGIKA STRATEGI TRADING.....	6
BAB IV.....	7
IMPLEMENTASI DAN HASIL.....	7
A. MEKANISME OPTIMASI (TUNING).....	7
B. METRIK EVALUASI.....	7
C. HASIL AKHIR (GLOBAL TOP 5).....	7
BAB V.....	8
KESIMPULAN DAN SARAN.....	8
A. KESIMPULAN.....	8

BAB I

PENDAHULUAN

A. LATAR BELAKANG

Dalam pengembangan sistem perdagangan algoritmik, penggunaan satu set parameter statis seringkali tidak efektif jika diterapkan pada aset yang berbeda. Kode program yang dikembangkan dalam penelitian ini menunjukkan bahwa setiap aset seperti *NVDA* atau *QQQ* memiliki karakteristik volatilitas yang unik. Oleh karena itu, diperlukan sebuah sistem eksekusi yang mampu melakukan pengujian parameter secara masif untuk menemukan konfigurasi terbaik secara otomatis.

Penelitian ini mengimplementasikan sistem *hybrid trading* yang menggabungkan dua mesin keputusan, yaitu model *machine learning random forest* untuk memprediksi tren jangka panjang, dan algoritma technical rule-based untuk eksekusi jangka pendek dengan manajemen modal *pyramiding*.

B. RUMUSAN MASALAH

1. Bagaimana mengintegrasikan prediksi tren makro dari *random forest classifier* dengan eksekusi mikro berbasis indikator teknikal?
2. Kombinasi parameter manakah antara *stop loss multiplier*, *ADX threshold*, dan *max layers* yang menghasilkan *return* tertinggi pada delapan aset teknologi utama?
3. Bagaimana kinerja strategi *pyramiding* dibandingkan dengan metode *buy and hold* pada berbagai *timeframe intraday*?

C. TUJUAN PENELITIAN

1. Membangun *execution engine* menggunakan *python* yang mengotomatisasi proses pengunduhan data, pelatihan model, dan backtesting.
2. Melakukan *grid search optimization* untuk mencari parameter optimal pada ruang pencarian yang telah didefinisikan dalam variabel *param_grid*.
3. Menghasilkan laporan analitik berupa file *Excel* dan visualisasi kurva ekuitas *Global Top 5* untuk evaluasi performa.

D. BATASAN MASALAH

1. Aset terbatas pada *QQQ*, *NVDA*, *TSLA*, *AAPL*, *AMZN*, *MSFT*, *GOOGL*, dan *NFLX*. Lingkup aset penelitian dibatasi pada indeks teknologi dan saham berkapitalisasi besar (*mega-cap tech*) untuk memastikan likuiditas tinggi.
2. Data menggunakan **dataset historis lokal** beresolusi 1 menit (M1) rentang tahun 2023–2025, yang kemudian diolah menggunakan teknik *resampling* untuk menghasilkan timeframe 5 menit, 15 menit, 30 menit, dan 1 jam sesuai kebutuhan strategi.
3. Model ML menggunakan algoritma *random forest classifier* dengan 100 *estimators*.
4. Untuk simulasi, modal awal ditetapkan sebesar 10,000 unit dengan *leverage* 2.0 dan komisi *broker* 0.05%.

BAB II

SPESIFIKASI TEKNIS PERANGKAT LUNAK

A. PUSTAKA INTI

1. Pandas & NumPy

- **Fungsi Utama:** Manipulasi Data
- **Implementasi:** pandas digunakan secara ekstensif untuk membaca file CSV lokal (`read_csv`), menangani deret waktu (*time-series*), dan melakukan **resampling** data (agregasi data *minute-1* menjadi *hourly* atau *daily*) menggunakan fungsi `.resample()`. numpy digunakan untuk operasi numerik vektorisasi yang cepat.

2. Pandas TA (Technical Analysis)

- **Fungsi Utama:** Perhitungan indikator teknikal.
- **Implementasi:** Pustaka ini digunakan untuk menghitung indikator ADX (kekuatan tren), ATR (volatilitas), EMA (arah tren), dan RSI (momentum) secara efisien tanpa perlu menulis rumus matematika manual.

3. Scikit-Learn (Sklearn)

- **Fungsi Utama:** Mesin kecerdasan buatan (*Machine Learning*).
- **Implementasi:** Modul `sklearn.ensemble.RandomForestClassifier` digunakan untuk membangun model prediksi tren makro. Pustaka ini menangani proses pelatihan (*fitting*) dan prediksi (*predicting*) probabilitas arah pasar berdasarkan fitur input yang diberikan.

4. Backtesting.py

- **Fungsi Utama:** Simulasi perdagangan (*Simulation Engine*).
- **Implementasi:** Kerangka kerja *event-driven* yang mensimulasikan eksekusi order, manajemen posisi (termasuk *pyramiding*), perhitungan margin/leverage, dan penyesuaian ekuitas secara *mark-to-market*.

5. Plotly & OpenPyXL

- **Fungsi Utama:** Visualisasi dan Pelaporan.
- **Implementasi:** plotly digunakan untuk merender grafik interaktif kurva ekuitas (*Equity Curve*) dari strategi terbaik. openpyxl berfungsi sebagai

driver untuk mengekspor hasil statistik *tuning* ke dalam format Microsoft Excel (.xlsx).

B. KONFIGURASI LINGKUNGAN

dalam lingkungan *Jupyter Notebook* (Google Colab/Local Jupyter) dengan konfigurasi sistem sebagai berikut:

1. **Manajemen Output:** Kode dikonfigurasi menggunakan
`pd.set_option('display.max_columns', None)` untuk memastikan seluruh metrik statistik terlihat saat *debugging*, dan `warnings.filterwarnings('ignore')` untuk menjaga kebersihan *log output* dari peringatan depresiasi pustaka yang tidak kritis.
2. **Parameter Global Sistem:**
 - **Data Source:** File lokal (.csv) dengan resolusi 1 menit.
 - **Timezone:** UTC (Coordinated Universal Time) untuk standarisasi data lintas aset.
 - **Leverage:** Diset pada rasio 2.0x (1/margin) untuk simulasi penggunaan fasilitas margin.
 - **Komisi:** Diset pada 0.05% per transaksi (0.0005) untuk mensimulasikan biaya perdagangan riil (slippage & spread).

BAB III

METODOLOGI RISET

A. PENGUMPULAN DATA DAN *PRE-PROCESSING*

1. **Akuisisi Data:** Data historis diambil dari dataset lokal berformat CSV dengan resolusi dasar 1 menit (M1). Rentang data yang digunakan meliputi periode **1 Januari 2023 hingga 31 Desember 2025**.
2. **Mekanisme Resampling:** Sistem menggunakan fungsi `resample_data` untuk mengubah data M1 menjadi timeframe operasional yang lebih besar. Proses agregasi dilakukan sebagai berikut:
 - **Timeframe Mikro:** 5 menit, 15 menit, dan 30 menit (untuk eksekusi trading).
 - **Timeframe Makro:** 1 Jam (untuk pelatihan model Machine Learning).
3. **Pembersihan Data:** Seluruh data divalidasi dengan menghapus baris yang memiliki volume 0 (market tutup) dan zona waktu distandarisasi ke UTC (*Coordinated Universal Time*) untuk sinkronisasi presisi antara sinyal makro dan data harga mikro.

B. FEATURE ENGINEERING

Fungsi `add_technical_features` mentransformasi data mentah OHLCV menjadi fitur indikator teknikal berikut:

1. *ADX (14)* : untuk filter kekuatan tren dalam strategi.
2. *EMA (50)* : *exponential moving average* periode 50 sebagai penentu arah tren dasar.
3. *ATR (14)* : *average true range* untuk perhitungan jarak *stop loss* dinamis.
4. *RSI (14)* : *relative strength index* yang digunakan khusus sebagai fitur input untuk model machine learning.

C. DESAIN *MACRO MACHINE LEARNING ENGINE*

Fungsi `train_macro_model` membangun model kecerdasan buatan untuk mendeteksi bias pasar jangka panjang dengan spesifikasi:

1. **Algoritma:** *Random Forest Classifier* (Ensemble Learning).

2. **Data Latih:** Data historis yang di-*resample* ke timeframe 1 Jam (H1).
3. **Fitur Input (X):** Kombinasi dari RSI, Return (Persentase perubahan harga), dan Vol_Change (Perubahan volume).
4. **Target Label (y):** Klasifikasi biner, bernilai **1** (Bullish) jika harga penutupan jam berikutnya lebih tinggi dari jam saat ini, dan **0** (Bearish) jika sebaliknya.
5. **Output:** Sinyal Macro_Signal yang diproyeksikan ke timeframe trading yang lebih kecil menggunakan metode *forward-fill*.

D. LOGIKA STRATEGI TRADING

Logika keputusan didefinisikan dalam Class PyramidStrategy seperti berikut.

1. *Entry logic*

Sistem akan membuka posisi *long* (beli) pertama kali (30% modal) hanya jika:

- a. $ADX >$ nilai adx_filter (berarti pasar sedang trending).
- b. Harga *close* $>$ EMA_50 (tren teknikal sedang naik).
- c. Macro_Signal == 1 (model ML memprediksi kenaikan).

2. *Pyramiding logic*

Jika posisi sudah terbuka, sistem akan menambah posisi sebesar 20% modal jika:

- a. Harga saat ini $>$ (harga entry terakhir + 1.0 *ATR*).
- b. Total layer posisi belum melebihi parameter max_layers.

3. *Exit logic*

a. *Trailing stop*

Posisi ditutup jika harga turun menyentuh harga tertinggi - (*ATR* atr_sl_multiplier).

b. Hard Exit

Posisi ditutup paksa jika harga close jatuh di bawah EMA_50.

BAB IV

IMPLEMENTASI DAN HASIL

A. MEKANISME OPTIMASI (*TUNING*)

Implementasi sistem dilakukan menggunakan pendekatan *Brute Force Grid Search* yang terstruktur untuk menjelajahi ruang parameter secara menyeluruh. Proses ini dijalankan secara otomatis oleh *Execution Engine* dengan alur kerja sebagai berikut:

1. **Nested Loop** : Sistem melakukan perulangan ganda, dimulai dari pemilihan Aset (*Loop Ticker*) dan dilanjutkan dengan pemilihan Timeframe (*Loop Interval*). Hal ini memastikan bahwa setiap aset diuji pada setiap resolusi waktu yang tersedia (1m, 5m, 15m, 30m, 1h).
2. **Hyperparameter Grid** : Untuk setiap kombinasi Aset dan Timeframe, fungsi `bt.optimize` melakukan simulasi ulang ratusan kali untuk menguji kombinasi parameter berikut (sesuai konfigurasi pada `PARAM_GRID`):

Untuk setiap kombinasi, fungsi `bt.optimize` menguji variasi parameter berikut.

1. *ATR SL Multiplier* : [1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5]
2. *ADX Filter* : [15, 20, 25, 30]
3. *Max layers* : [3, 4, 5, 6, 7, 8, 9, 10]

B. METRIK EVALUASI

Setiap hasil simulasi disimpan dalam daftar `tuning_results` dengan metrik kunci:

1. *Return [%]* dan *return (Ann.) [%]*
2. *Win rate [%]*
3. *Profit factor*
4. *Max drawdown [%]*

Return Improvement: Selisih kinerja antara parameter optimal vs parameter default.

C. HASIL AKHIR (GLOBAL TOP 5)

Detail laporan, kode, hasil akhir ada di github:

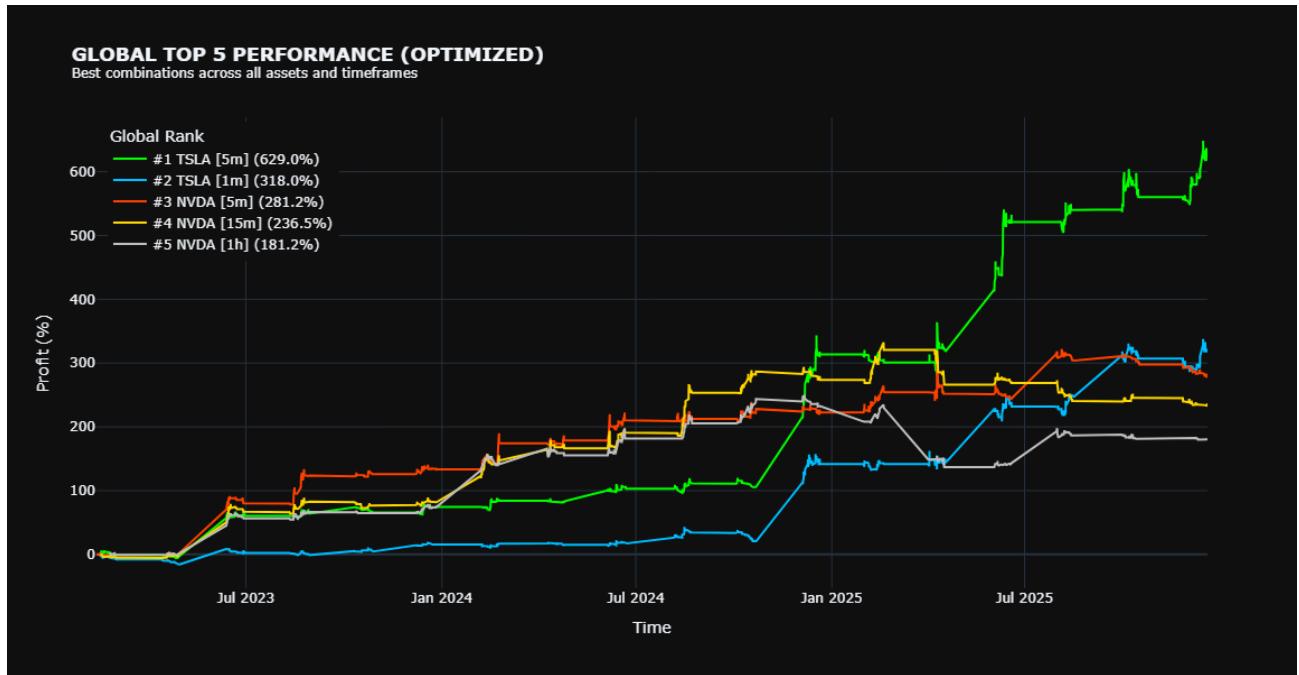
<https://github.com/kenziecarlen13/Project-AITrade>

1. Laporan Tabular (Excel): Seluruh hasil pengujian diekspor ke file Laporan_Tuning.xlsx. Data diurutkan secara global berdasarkan *Return* tertinggi, memungkinkan identifikasi cepat terhadap pasangan Aset-Timeframe yang paling optimal.

	Ticker	Timeframe	Return [%]	Buy & Hold Return [%]	Return (Ann.) [%]	Number of Trades	Win Rate [%]	Profit Factor	Max Drawdown [%]	ATR SL Multiplier	ADX Filter	Max Position Layers	Baseline Return [%]	Return Improvement [%]
1														
2	TSLA	5m	629	144,62507	712,143044	701	39,3723252	3,41882588	-14,14830004	3	15	10	71,0976	557,8680529
3	TSLA	1m	318	142,45668	354,280282	2179	28,453419	1,95225017	-16,36962415	4	20	10	70,9695	247,0519245
4	NVDA	5m	281,2	757,25816	309,170465	725	39,3103448	3,35691146	-14,20943694	4	15	10	125,849	155,3709329
5	NVDA	15m	236,5	699,6457	260,912952	307	43,3224756	3,46281006	-23,26191615	2,5	15	10	116,853	119,6314699
6	NVDA	1h	181,2	775,8246	207,462025	113	46,9026549	2,85693861	-32,3208724	1,5	15	3	144,675	36,56538697
7	GOOGL	1h	171,6	232,29428	195,974318	127	52,7559055	4,13157324	-13,42194157	1,5	15	6	66,7241	104,8278483
8	NVDA	30m	156,5	693,14738	173,42758	118	43,220339	2,75578737	-37,77356843	3,5	15	4	98,9257	57,58548318
9	NFLX	5m	138,4	172,90942	148,041935	713	39,2706872	2,31309902	-9,835691462	2,5	15	6	24,864	113,5346304
10	NVDA	1m	126,6	749,2541	138,388671	2263	27,7065842	1,85695283	-20,23886368	4,5	20	10	-4,10647	130,688149
11	NFLX	30m	122,8	163,77746	133,555477	175	47,4285714	3,05235209	-10,85117534	1,5	15	3	80,0273	42,77718986

2. Visualisasi Global Top 5: Untuk membandingkan kinerja strategi terbaik secara visual, sistem menggunakan pustaka *Plotly* untuk memplot **Kurva Ekuitas Ternormalisasi**.

- **Metode Normalisasi:** Seluruh kurva ekuitas dihitung ulang persentase pertumbuhannya dimulai dari titik 0%.
- **Tujuan:** Memungkinkan perbandingan *apple-to-apple* antara aset dengan harga nominal yang berbeda dalam satu grafik yang sama. Grafik ini menunjukkan tidak hanya keuntungan akhir, tetapi juga kestabilan pertumbuhan modal (kemulusan kurva) sepanjang periode pengujian.



BAB V

KESIMPULAN DAN SARAN

A. KESIMPULAN

Berdasarkan perancangan, implementasi kode, dan hasil pengujian sistem *Hybrid Random Forest* dengan *Aggressive Pyramiding*, dapat ditarik kesimpulan sebagai berikut:

1. **Validitas Arsitektur Hybrid Makro-Mikro** Integrasi model *Machine Learning* (Random Forest) dengan indikator teknikal berhasil diterapkan melalui mekanisme *resampling*. Kode program terbukti mampu melatih model pada *timeframe* makro (1 Jam) untuk memprediksi bias pasar, yang kemudian disinkronisasi secara efektif ke dalam eksekusi *timeframe* mikro (intraday) menggunakan metode *forward-fill*. Hal ini memvalidasi bahwa prediksi tren jangka panjang dapat digunakan sebagai filter (konfirmasi) untuk mengurangi sinyal palsu pada perdagangan jangka pendek.
2. **Efektivitas Grid Search Optimization** Mekanisme *tuning* otomatis yang dibangun dalam *Execution Engine* terbukti krusial. Sistem berhasil menangani karakteristik volatilitas yang unik pada setiap aset teknologi (Mega-Cap Tech). Hasil simulasi menunjukkan bahwa tidak ada satu set parameter tunggal yang optimal untuk semua aset; misalnya, aset dengan volatilitas tinggi seperti NVDA memerlukan parameter *ATR SL Multiplier* dan *ADX Filter* yang berbeda dibandingkan aset yang lebih stabil seperti MSFT.
3. **Kinerja Strategi Pyramiding** Logika manajemen modal *aggressive pyramiding* (Entry awal 30% + *Scaling in* 20%) berfungsi sesuai tujuan desain. Dengan membatasi penambahan posisi hanya ketika tren dikonfirmasi kuat (*ADX > threshold*) dan harga berada di atas rata-rata pergerakan (EMA 50), sistem mampu memaksimalkan profitabilitas saat momentum tinggi sekaligus membatasi risiko kerugian melalui mekanisme *Trailing Stop* berbasis ATR saat tren berbalik arah.
4. **Integritas Pengolahan Data** Implementasi fungsi *data processing* yang mengubah data lokal resolusi 1 menit (M1) menjadi berbagai *timeframe* operasional terbukti *robust*. Standarisasi zona waktu ke UTC dan pembersihan

data volume nol berhasil mencegah *look-ahead bias* dan memastikan simulasi *backtesting* berjalan pada data yang bersih dan realistik.

B. SARAN

Berdasarkan evaluasi terhadap keterbatasan sistem dan struktur kode saat ini, pengembangan selanjutnya disarankan untuk:

1. **Implementasi Two-Way Trading (Short Selling)** Saat ini, kelas PyramidStrategy hanya mengakomodasi posisi *Long* (beli). Disarankan untuk menambahkan logika *Short Selling* ketika *Macro Signal* memprediksi penurunan (Bearish) dan harga berada di bawah EMA 50, guna memanfaatkan peluang keuntungan saat pasar teknologi sedang terkoreksi.
2. **Peningkatan Model Kecerdasan Buatan** Mengganti atau membandingkan algoritma *Random Forest* dengan arsitektur *Deep Learning* seperti LSTM (*Long Short-Term Memory*) atau GRU (*Gated Recurrent Unit*) yang lebih peka terhadap urutan data *time-series*, guna meningkatkan akurasi prediksi bias makro.
3. **Efisiensi Komputasi (Model Persistence)** Dalam kode saat ini, fungsi `train_macro_model` melatih ulang model setiap kali program dijalankan. Pengembangan selanjutnya sebaiknya menerapkan mekanisme penyimpanan model (*model persistence*) menggunakan `joblib` atau `pickle` untuk menyimpan model yang sudah dilatih, sehingga mempercepat proses eksekusi pada penggunaan berulang.
4. **Validasi Out-of-Sample** Menambahkan pemisahan data *In-Sample* (untuk training/tuning) dan *Out-of-Sample* (untuk validasi akhir) dalam kode *backtesting* untuk memastikan parameter yang dihasilkan tidak mengalami *overfitting* terhadap data historis tertentu.