Arus Jurnal Sains dan Teknologi (AJST)
Website: http://jurnal.ardenjaya.com/index.php/ajst
Email: jurnal.ajst@ardenjaya.com

Optimasi Portofolio Saham Berbasis Metaheuristik dengan Pemilihan Saham melalui K-Means, Fuzzy C-Means, dan Agglomerative Clustering serta Analisis Return melalui Simulasi Latent Performance Investment

INFO PENPULIS

Alfiyya Dzawi Rahmah
Universitas Gadjah Mada
alfiyyadzawir@gmail.com

Anggita Dwi Putri
Universitas Gadjah Mada
anggitap638@gmail.com

Farsya Alaika
Universitas Gadjah Mada
farsyaalaikaaa@gmail.com

Nabila Alayya Salzabil Lukito
Universitas Gadjah Mada
nabilaalayya23@gmail.com

Abdurrakhman
Universitas Gadjah Mada
rakhmanstat@ugm.ac.id

INFO ARTIKEL

ISSN: 3026-3603
Vol. 4, No. 2 Oktober 2026
http://jurnal.ardenjaya.com/index.php/ajst

© 2026 Arden Jaya Publisher All rights reserved

Saran Penulisan Referensi:

Rahmah, A. D., Putri, A. D., Alaika, F., Lukito, N. A. S., & Abdurrakhman. (2026). Optimasi Portofolio Saham Berbasis Metaheuristik dengan Pemilihan Saham melalui K-Means, Fuzzy C-Means, dan Agglomerative Clustering serta Analisis Return melalui Simulasi Latent Performance Investment. *Arus Jurnal Sains dan Teknologi , 4* (2), 539-559.

Abstrak

Pasar modal berperan penting sebagai sarana pendanaan bagi perusahaan, sekaligus wadah investasi bagi masyarakat. Namun, fluktuasi harga saham yang tinggi menimbulkan risiko yang perlu dikelola secara cermat. Analisis ini bertujuan membangun portofolio saham yang efisien melalui integrasi metode clustering dan algoritma optimisasi metaheuristik. Sebanyak 25 saham dari lima sektor utama di Bursa Efek Indonesia (BEI) digunakan sebagai objek penelitian dengan periode pengamatan 2015–2019. Proses pengelompokan saham dilakukan menggunakan tiga metode clustering, yaitu K-Means, Fuzzy C-Means, dan Agglomerative Clustering. Hasil analisis menunjukkan bahwa metode K-Means memberikan performa terbaik dengan nilai silhouette score tertinggi, yaitu sebesar 0,4333. Tahap optimasi selanjutnya dilakukan menggunakan delapan algoritma metaheuristik dengan Grey Wolf Optimization (GWO) menghasilkan kinerja paling unggul dengan sharpe ratio tertinggi, yaitu sebesar 1,392 serta risiko terendah berdasarkan value at risk dan probability of loss. Simulasi investasi sebesar Rp500.000.000 pada akhir tahun 2019 menunjukkan pertumbuhan nilai portofolio sebesar 46,6% dengan tingkat pengembalian tahunan majemuk (CAGR) 7,95%. Hasil ini membuktikan bahwa kombinasi klasterisasi K-Means dan optimasi berbasis GWO efektif dalam membentuk portofolio efisien, stabil, dan relevan bagi strategi investasi pasif di pasar saham Indonesia.

Abstract

The capital market plays a vital role as a funding source for companies and an investment vehicle for the public. However, high stock price volatility introduces risks that must be carefully managed. This study aims to construct an efficient stock portfolio by integrating clustering methods with metaheuristic optimization algorithms. A total of 25 stocks from five major sectors listed on the Indonesia Stock Exchange (IDX) were used as research objects, covering the observation period from 2015 to 2019. The stock grouping process was carried out using three clustering methods: K-Means, Fuzzy C-Means, and Agglomerative Clustering. The analysis results indicate that the K-Means method achieved the best performance with the highest silhouette score of 0.4333. The subsequent optimization stage employed eight metaheuristic algorithms, among which the Grey Wolf Optimization (GWO) algorithm outperformed others by producing the highest Sharpe ratio of 1.392 and the lowest risk based on value at risk (VaR) and probability of loss. An investment simulation of IDR 500,000,000 at the end of 2019 demonstrated a portfolio value growth of 46.6% with a compound annual growth rate (CAGR) of 7.95%. These findings confirm that the combination of K-Means clustering and GWO-based optimization is effective in forming an efficient, stable, and relevant portfolio for passive investment strategies in the Indonesian stock market.

Keywords: Stocks, Clustering, Portfolio Optimization, Metaheuristic Methods, Simulation, Latent Performance Investment

Kata Kunci: Saham, Clustering, Optimasi Portofolio, Metode Metaheuristik, Simulasi, Latent Performance Investment

A. Pendahuluan

Pasar modal merupakan salah satu pilar utama dalam sistem keuangan moderen karena menyediakan sarana bagi perusahaan untuk memperoleh pendanaan, sekaligus memberikan peluang investasi bagi investor (Levine, 2011). Perkembangan pesat pasar modal di Indonesia dalam dua dekade terakhir telah meningkatkan partisipasi investor ritel dan institusi. Namun, dinamika pasar yang tinggi juga membawa tantangan berupa risiko volatilitas harga saham yang tidak dapat diabaikan. Dalam konteks ini, manajemen risiko dan strategi optimasi portofolio menjadi aspek penting untuk mencapai imbal hasil yang optimal dengan tingkat risiko yang terkendali. Melalui teori portofolio moderen, Markowitz memperkenalkan konsep *mean-variance optimization* yang menjadi tonggak utama dalam pengelolaan portofolio. Akan tetapi, seiring perkembangan pasar keuangan dan meningkatnya kompleksitas data, berbagai studi mutakhir menunjukkan bahwa pendekatan klasik ini memiliki keterbatasan, terutama ketika dihadapkan pada permasalahan berskala besar dan non-linear (Erwin & Engelbrecht, 2023).

Dalam praktiknya, salah satu tantangan terbesar dalam pembentukan portofolio adalah pemilihan aset yang tepat sebelum dilakukan optimasi bobot. Pemilihan saham tidak hanya berdasarkan *return* historis semata, tetapi juga perlu mempertimbangkan karakteristik risiko dan pola pergerakan harga yang serupa. Metode *clustering* banyak digunakan dalam literatur keuangan sebagai pendekatan untuk mengelompokkan saham dengan perilaku yang mirip sehingga memudahkan proses seleksi awal portofolio.

Namun, proses optimasi bobot portofolio tetap menjadi tahap yang sangat penting. Berbagai metode klasik, seperti *mean-variance* dari teori Markowitz sering kali tidak mampu memberikan solusi optimal ketika berhadapan dengan banyak kendala praktis, misalnya batasan jumlah aset, non-linearitas hubungan antar variabel, dan data *return* yang tidak selalu berdistribusi normal. Untuk mengatasi hal ini, pendekatan metaheuristik mulai banyak diterapkan karena kemampuannya dalam menangani masalah optimasi kompleks dan berskala besar (Boussaïd et al., 2013).

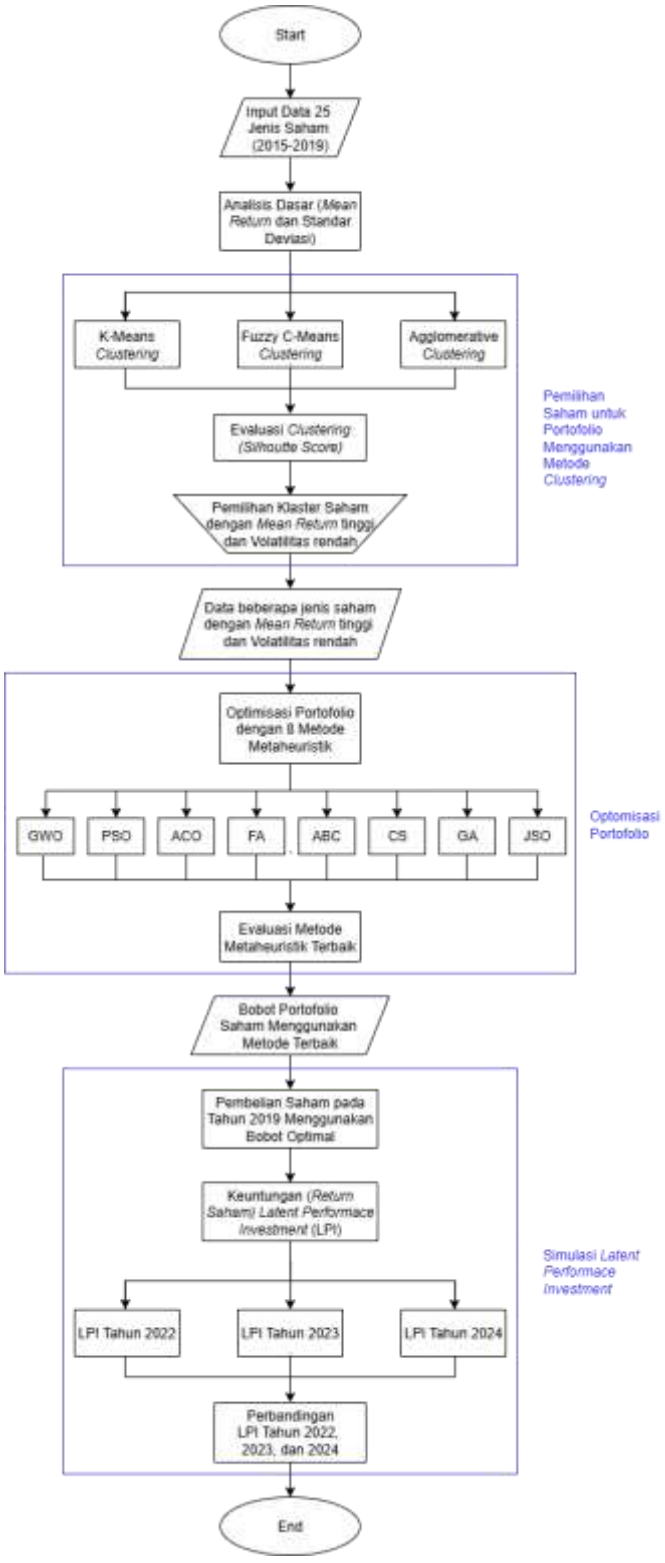
Urgensi analisis ini juga diperkuat oleh semakin dinamisnya kondisi pasar saham Indonesia yang dipengaruhi faktor makroekonomi global, seperti inflasi, suku bunga, hingga volatilitas pasar komoditas. Dalam kondisi demikian, investor membutuhkan pendekatan kuantitatif yang dapat beradaptasi dengan cepat terhadap perubahan struktur data. Analisis ini menjelaskan integrasi antara teknik *clustering* dan optimasi metaheuristik sebagai solusi praktis dan inovatif untuk menghasilkan portofolio dengan kinerja lebih baik.

Berdasarkan kondisi tersebut, analisis ini diarahkan untuk memberikan kontribusi melalui fokus kajian pada aspek-aspek berikut.

a. Konsistensi pemilihan saham lintas sektor dengan karakteristik *return* dan volatilitas dalam Bursa Efek Indonesia (BEI).
b. Perbandingan efektivitas beberapa metode *clustering* (K-Means, Fuzzy C-Means, Agglomerative) dalam seleksi saham dengan kriteria efisiensi.
c. Aplikasi beberapa algoritma metaheuristik dalam optimasi bobot portofolio.
d. Evaluasi menyeluruh performa portofolio melalui simulasi, menggunakan metrik lengkap agar investor ataupun akademisi dapat melihat implikasi praktisnya.

B. Metodologi

Tahapan Analisis



Gambar 1. *Flowchart* Analisi

Saham

Saham adalah salah satu instrumen investasi yang paling populer di pasar modal. Instrumen ini mencerminkan kepemilikan atas suatu perusahaan sehingga pemegang saham berhak menerima dividen, potensi kenaikan harga saham, serta klaim atas aset perusahaan apabila terjadi likuidasi. Selain itu, saham dapat diperjualbelikan dengan relatif mudah di bursa sehingga menjadi instrumen yang likuid dan menarik bagi banyak investor.

Bagi investor, ukuran yang paling penting dari sebuah saham adalah *return*, yaitu imbal hasil yang diperoleh dari investasi dalam periode tertentu. Salah satu cara yang umum digunakan untuk mengukur return adalah *log-return* karena mampu menggambarkan perubahan harga secara proporsional.

Return

Return dalam penelitian portofolio merujuk pada tingkat keuntungan yang diperoleh investor dari suatu aset atau portofolio selama periode tertentu. *Return* biasanya dibedakan menjadi dua bentuk *return* realisasi (*realized return*) yang dihitung dari data historis dan *return* ekspektasi (*expected return*) yang digunakan sebagai proyeksi untuk keputusan investasi di masa depan (Bodie et al., 2014).

Dalam analisis ini, *return* yang digunakan adalah *return* harian saham yang dihitung menggunakan *logarithmic return* untuk menjaga sifat aditif pada periode waktu yang berbeda serta mengurangi distorsi akibat fluktuasi harga yang besar. Rumusnya dapat dituliskan sebagai berikut.

$$R_{i(t)} = \ln\left(\frac{P_t}{P_{t-1}}\right) \tag{1}$$

dengan

$R_{i(t)}$: *return* saham ke-i pada periode ke-t,
P_t : harga saham pada periode ke-t,
P_{t-1} : harga saham pada periode sebelumya.

Dalam analisis ini, harga penutupan (*closing price*) digunakan sebagai dasar perhitungan *return*.

Dua properti penting dari *return* saham yang digunakan dalam analisis ini adalah *mean return* dan volatilitas. *Mean return* merepresentasikan rata-rata imbal hasil yang diharapkan dari suatu saham yang secara matematis dituliskan sebagai berikut.

$$E[R_i] = \frac{1}{T}\sum_{t=1}^{T} R_i\,(t) \tag{2}$$

dengan

$E[R_i]$: *expected return* saham ke- *i,*
T : jumlah periode pengamatan.

Volatilitas

Volatilitas atau risiko saham menggambarkan tingkat ketidakstabilan atau fluktuasi *return* yang terjadi selama periode tertentu. Semakin tinggi volatilitas, semakin besar pula risiko yang harus ditanggung investor akibat perubahan harga saham yang tidak menentu. Dalam konteks kuantitatif, volatilitas umumnya diukur menggunakan standar deviasi dari return historis karena standar deviasi mencerminkan seberapa jauh nilai *return* menyimpang dari rata-rata ekspektasinya.

Rumus perhitungannya dapat dinyatakan sebagai berikut.

$$\sigma_i = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(R_i(t) - E[R_i])^2} \tag{3}$$

dengan

σ_i : volatilitas (standar deviasi) saham ke- *i,*
$R_{i(t)}$: *return* saham ke-i pada periode ke-t,
$E[R_i]$: *expected return* saham ke- *i,*
T : jumlah periode pengamatan.

Portofolio

Teori portofolio pertama kali diperkenalkan oleh Harry M. Markowitz pada tahun 1952. Secara umum, portofolio merupakan proses pengelolaan berbagai aset investasi dengan tujuan mencapai imbal hasil yang optimal sesuai dengan profil risiko investor (Abdurakhman et al., 2025). Konsep portofolio efisien mengacu pada portofolio yang mampu memberikan *return* ekspektasi tertinggi dengan tingat risiko yang sudah pasti atau sebaliknya. Artinya, sebuah portofolio dikatakan efisien apabila memiliki tingkat risiko yang sama dan juga mampu memberikan tingkat keuntungan yang lebih tinggi atau mampu menghasilkan tingkat keuntungan yang sama, tetapi dengan risiko yang lebih rendah (Nurlaeli & Artati, 2020).

Sementara itu, portofolio optimal adalah portofolio yang dipilih antara portofolio efisien berdasarkan preferensi risiko investor dan ujuan investasi sehingga menghasilkan keseimbangan terbaik antara *return* dan risiko sesuai toleransi investor (Sarjana et al., 2022). Dengan demikian, portofolio optimal adalah portofolio yang memberikan hasil kombinasi *return* tertinggi dengan risiko atau volatilitas terendah. *Return* portofolio sendiri merupakan total imbal hasil yang didapatkan dari keseluruhan aset dalam portofolio yang dihitung menggunakan rumus sebagai berikut.

$$Return\ Portofolio = \sum (w_i \times r_i) \tag{4}$$

dengan

w_i : proporsi investasi pada aset ke-i,

r_i : *return* aset tersebut.

Risiko portofolio diperkenalkan oleh Harry M. Markowitz pada tahun 1950. Risiko ini mengacu pada volatilitas atau simpangan baku dari *return* portofolio yang menggambarkan ketidakpastian hasil investasi. Salah satu prinsip penting dari teori portofolio adalah bahwa risiko dapat dikurangi dengan menggabungkan beberapa sekuritas tunggal menjadi sebuah portofolio sehingga diversifikasi dapat menurunkan kemungkinan kerugian tanpa harus mengorbankan potensi *return* secara signifikan.

K-Means *Clustering*

K-Means merupakan salah satu metode *clustering* nonhierarki yang berusaha untuk mengelompokkan objek berdasarkan karakteristiknya. Objek-objek yang memiliki karakteristik serupa akan dikelompokkan ke dalam satu *cluster*, sedangkan objek dengan karakteristik berbeda akan ditempatkan pada *cluster* lain (Musharyadi, 2017). Algoritma ini sangat populer karena kesederhanaannya, kemampuan untuk menangani data yang besar, serta efisiensi dalam menghadapi data *outlier.* Tahapan pada algoritma K-Means adalah sebagai berikut.

1) Menentukan jumlah *cluster* yang akan dibentuk.
2) Membangkitkan titik pusat *cluster* awal (*centroid*) secara acak. Jumlah *centroid* yang terbentuk sesuai dengan jumlah *cluster* yang akan dibentuk.
3) Hitung jarak antara objek dengan *centroid* menggunakan persamaan *Euclidean Distance*. *Euclidean Distance* adalah perhitungan untuk menentukan jarak dari dua buah titik. Persamaan *Euclidean Distance* adalah sebagai berikut.

$$d_{ij} = \sqrt{\sum_{k=1}^{p} \left(x_{ij} - x_{jk}\right)^2} \tag{5}$$

dengan

d_{ij} : jarak antar objek *i* dengan *j,*

p : dimensi data,

x_{ij} : koordinat objek *i* pada dimensi *k,*

x_{jk} : koordinat objek *j* pada dimensi *k.*

4) Mengelompokkan setiap data ke *cluster* yang memiliki jarak terdekat dengan titik *centroid.*
5) Melakukan pengulangan (iterasi) dari membangkitkan nilai *centroid* yang baru diperoleh dengan persamaan berikut.

$$C_k = \frac{1}{n_k} \sum d_i \tag{6}$$

dengan

n_k : jumlah data dalam *cluster,*

d_i : jumlah dari nilai jarak yang masuk dalam masing-masing *cluster.*

6) Melakukan iterasi hingga didapatkan nilai *centroid* dan anggota masing-masing *cluster* tidak berubah. Jika kondisi ini terpenuhi, didapatkan hasil *cluster* final, tetapi jika kondisi ini tidak terpenuhi, dilakukan pengulangan kembali dari langkah (2).

Fuzzy C-Means *Clustering*

Fuzzy C-Means (FCM) adalah metode *clustering* yang diperkenalkan pertama kali oleh Jim Bezdek pada tahun 1981. Metode ini menetapkan derajat keanggotaan suatu titik pada setiap *cluster* sehingga suatu titik data dapat menjadi anggota lebih dari satu *cluster* dengan bobot tertentu. FCM memanfaatkan konsep himpunan *fuzzy* sebagai dasar pembobotan suatu *cluster* atau kelompok (Martin & Nataliani, 2020).

Konsep dasar FCM dimulai dengan menentukan pusat *cluster* yang merepresentasikan lokasi rata-rata setiap *cluster*. Setiap titik data diberikan derajat keanggotaan pada setiap cluster. Dengan memperbarui pusat *cluster* dan derajat kenggotaan secara berulang, posisi pusat *cluster* akan bergerak menuju lokasi yang paling representatif. Proses iteratif ini didasarkan pada minimisasi fungsi objektif yang menggambarkan jarak dari titik data dan pusat *cluster* yang terbobot oleh derajat keanggotaan masing-masing titik data (Putri et al., 2022). Tahapan pada algoritma Fuzzy C-Means adalah sebagai berikut.

1) Memasukkan data (X_{ij}) yang akan di-*cluster* ke dalam sebuah matriks berukuran $n \times m$, dengan n merepresentasikan jumlah pengamatan dan m adalah atribut setiap data.
2) Menetapkan parameter *fuzzy clustering*, yaitu jumlah *cluster* (c) yang akan dibentuk, eksponen *fuzziness* (w), error terkecil yang diharapkan (ζ), fungsi objektif awal (P_0), maksimum iterasi, dan iterasi awal.
3) Membangkitkan bilangan random a_{ik}, antara 0–1 sebagai elemen matriks keanggotaan awal **U**.
4) Menghitung jumlah setiap baris pada matriks keanggotaan awal menggunakan persamaan berikut.

$$Q_i = \sum_{k=1}^{c} a_{ik} \tag{7}$$

dengan
Q_i : total jumlah nilai keanggotaan awal data ke-i terhadap seluruh *cluster,*
k : indeks *cluster.*

5) Normalisasi nilai keanggotaan awal menggunakan persamaan berikut.

$$\mu_{ik}^{(0)} = \frac{a_{ik}}{Q_i} \tag{8}$$

dengan
$\mu_{ik}^{(0)}$: nilai keanggotaan hasil normalisasi data dari ke-i terhadap *cluster* ke-k pada iterasi awal,
a_{ik} : nilai acak keanggotaan awal sebelum normalisasi,
Q_i : jumlah total baris untuk data ke-*i.*

6) Menghitung pusat *cluster* ke-k menggunakan persamaan berikut.

$$V_{jk} = \frac{\sum_{i=1}^{s}((\mu_{ik})^w X_{ij})}{\sum_{i=1}^{s}(\mu_{ik})^w} \tag{9}$$

dengan
V_{kj} : pusat *cluster* ke-k pada dimensi ke-*j,*
μ_{ik} : derajat keanggotaan data ke-i pada *cluster* ke-*k,*
w : eksponen *fuzziness,*
x_{ij} : nilai data ke-i pada dimensi ke-*j,*
s : jumlah data.

7) Menghitung fungsi objektif pada iterasi ke-t menggunakan persamaan berikut.

$$P_t = \sum_{i=1}^{s}\sum_{k=1}^{c}\left(\sum_{j=1}^{p} d_{ik}^2(x_{ij}, v_{kj})\right)(\mu_{ik})^w \tag{10}$$

dengan

P_t : nilai fungsi objektif pada iterasi ke-*t,*
$d_{ik}^2(x_{ij}, v_{kj})$: jarak kuadrat antara data ke-*i* dan pusat *cluster* ke-*k* pada dimensi ke-*j,*
μ_{ik} : derajat keanggotaan data ke-*i* terhadap *cluster* ke-*k,*
w : eksponen *fuzziness,*
p : jumlah dimensi,
k : jumlah *cluster.*

8) Menghitung perubahan matriks keanggotaan menggunakan persamaan berikut.

$$\mu_{ik}^{(0)} = \frac{a_{ik}}{Q_i} \tag{11}$$

dengan
$\mu_{ik}^{(t)}$: derajat keanggotaan data ke-*i* pada *cluster* ke-*k* pada iterasi ke-*t,*
$d_{ik}^2(x_{ij}, v_{kj})$: jarak kuadrat antara data ke-*i* dan pusat *cluster* ke-*k,*
w : eksponen *fuzziness.*

9) Mengecek kondisi berhenti, proses berhenti jika terpenuhi salah satu kondisi, yaitu jika $|P_t - P_{t-1}| < \zeta$ atau $t > MaxIter$. Apabila kedua syarat tersebut belum terpenuhi, perbarui nilai t menjadi $t + 1$, kemudian kembali ke langkah (6) untuk melanjutkan perhitungan.
10) Menentukan anggota pada setiap *cluster* dengan cara menyatukan objek data masuk ke dalam suatu *cluster* jika nilai keanggotaannya mendekati satu atau merupakan nilai yang paling besar dibandingkan dengan *cluster* lainnya.

Agglomerative Hierarchical *Clustering*

Agglomerative Hierarchical adalah metode *clustering* yang membentuk hierarki kelompok dengan menempatkan setiap data sebagai *cluster* individu (Handayani & Sitokdana, 2025). Metode ini bersifat *bottom-up,* yaitu menggabungkan *cluster* kecil secara bertahap hingga terbentuk satu *cluster* tunggal. Proses ini dimulai dari setiap data sebagai *cluster* awal. Kemudian, secara berulang dipilih pasangan *cluster* yang paling dekat untuk digabungkan menjadi *cluster* yang lebih besar. Proses ini dilakukan terus-menerus sehingga tampak bergerak ke atas membentuk struktur hierarki (Suhirman & Wintolo, 2019). Adapun algoritma yang digunakan dalam proses pengelompokan hierarki dengan metode Agglomerative antara lain adalah *single linkage, complete linkage, average linkage, ward method,* WPGMA *method, median method,* dan *centroid method* (Dani et al., 2019). Tahapan pada algoritma Agglomerative Hierarchical adalah sebagai berikut.

1) Menempatkan setiap objek data ke dalam satu *cluster* sendiri sehingga pada awalnya terdapat sebanyak *n cluster* (sama dengan jumlah objek data).
2) Menghitung jarak antar objek menggunakan persamaan Euclidean *Distance*. Persamaan Euclidean *Distance* adalah sebagai berikut.

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{ij} - x_{jk})^2} \tag{12}$$

dengan
d_{ij} : jarak antar objek *i* dengan *j,*
p : dimensi data,
x_{ij} : koordinat objek *i* pada dimensi *k.*

3) Menggabungkan *cluster* terdekat dengan cara identifikasi dua *cluster* yang memiliki jarak paling kecil dan gabungkan keduanya menjadi *cluster* baru.
4) Perbarui matriks jarak antar data untuk merepresentasikan kedekatan di antara kelompok baru dan kelompok yang masih tersisa.
5) Ulangi langkah (2) sampai (4) hingga hanya tersisa satu *cluster* yang mencakup seluruh objek data.

Silhouette Score

Silhouette merupakan metode evaluasi *cluster* yang menggabungkan metode *cohesion* dan *separation*. Cohesion diukur dengan menghitung jarak rata-rata setiap objek yang terdapat dalam sebuah *cluster* dan separation diukur dengan menghitung jarak rata-rata setiap objek dalam sebuah *cluster* dengan *cluster* terdekatnya (Kurniawan et al., 2023). Jumlah *cluster* terbaik adalah jumlah *cluster* yang memiliki rata-rata nilai *silhouette* tertinggi atau mendekati 1 (Dani et al., 2019). Persamaan rumus untuk menghitung *silhouette* adalah sebagai berikut.

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)} \quad (13)$$

dengan

a_i : rata-rata objek ke-*i* dengan semua objek pada satu kelompok yang sama,
b_i : rata-rata jarak objek ke-*i* dengan semua objek pada kelompok yang berbeda,
S_i : nilai koefisien *silhouette.*

Grey Wolf Optimization (GWO)

Algoritma Grey Wolf Optimizer (GWO) merupakan salah satu *nature-inspired algorithms* dalam kategori *swarm intelligence* yang diperkenalkan oleh Mirjalili et al. (Mirjalili et al., 2014). GWO terinspirasi dari perilaku sosial serigala abu-abu (*grey wolf*) dalam hal kepemimpinan, berburu, dan menyerang mangsa. Dibandingkan algoritma *swarm intelligence* lainnya, GWO dikenal sederhana, membutuhkan sedikit parameter, serta memiliki kinerja yang andal dalam berbagai permasalahan optimasi (Banaie-Dezfouli et al., 2021)–(Long et al., 2018).

Dalam algoritma GWO, populasi serigala abu-abu dibagi ke dalam empat tingkatan hierarki. Serigala alfa (α) merepresentasikan solusi terbaik, diikuti oleh beta (β) dan delta (δ) sebagai solusi terbaik kedua dan ketiga. Sementara itu, serigala lainnya ditempatkan dalam kategori omega (ω) yang berperan mengikuti arahan tiga pemimpin tersebut (Mirjalili et al., 2014), (Banaie-Dezfouli et al., 2021). Proses optimasi dalam GWO dijalankan melalui tiga mekanisme utama, yaitu melingkari mangsa, berburu mangsa, dan menyerang mangsa (Mirjalili et al., 2014), (Banaie-Dezfouli et al., 2021).

1) Melingkari Mangsa

Perilaku melingkari mangsa dimodelkan dengan:

$$D_i = \left|C_i \cdot X_p(t) - X_i(t)\right| \quad (14)$$

$$X_i(t+1) = X_p(t) - A_i \cdot D_i \quad (15)$$

A_i dan C_i dapat dihitung sebagai berikut.

$$A_i = 2a \cdot r_1 - a \quad (16)$$

$$C_i = 2 \cdot r_2 \quad (17)$$

$$a = 2 - \frac{2t}{t_{\max}} \quad (18)$$

dengan

D_i : jarak absolut antara posisi serigala ke-*i* dengan posisi mangsa,
X_i : posisi serigala ke-*i* pada iterasi ke-*t*,
X_p : posisi mangsa,
A_i : vektor kontrol yang mengatur jarak pergerakan,
C_i : vektor kontrol yang mengatur arah pergerakan,
$r_1, r_2 \in [0,1]$: bilangan acak *uniform*,
a : parameter yang menurun secara linier dari 2 ke 0 sesuai dengan iterasi.

2) Berburu Mangsa

Tiga serigala pemimpin (α, β, δ) berperan memperkirakan posisi mangsa. Posisi setiap serigala diperbarui berdasarkan ketiga pemimpin.

$$D_\alpha = \left|C_1 \cdot X_\alpha(t) - X_i(t)\right| \quad (19)$$

$$D_\beta = \left|C_2 \cdot X_\beta(t) - X_i(t)\right| \quad (20)$$

$$D_\delta = \left|C_3 \cdot X_\delta(t) - X_i(t)\right| \quad (21)$$

$$X_1 = X_\alpha(t) - A_1 \cdot D_\alpha \quad (22)$$

$$X_2 = X_\beta(t) - A_2 \cdot D_\beta \quad (23)$$

$$X_3 = X_\delta(t) - A_3 \cdot D_\delta \quad (24)$$

3) Menyerang Mangsa

Posisi serigala diperbarui dengan mengambil rata-rata ketiga posisi kandidat.

$$X_i(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (25)$$

Dengan mekanisme ini, serigala-serigala (ω) dipandu menuju posisi mangsa dengan arahan pemimpin (α, β, δ). Proses iterasi berlanjut hingga kriteria penghentian terpenuhi, misalnya jumlah iterasi maksimum atau tercapainya nilai fungsi objektif tertentu.

Particle Swarm Optimization (PSO)

Algoritma Particle Swarm Optimization (PSO) pertama kali diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995 sebagai metode optimasi stokastik berbasis populasi (Naiborhu

et al., 2013). PSO terinspirasi dari perilaku sosial kawanan burung dan serangga dalam mencari makanan, yaitu saat individu-individu dalam kelompok bergerak menuju posisi terbaik dengan mengikuti pengalaman pribadi maupun kelompok.

Dalam PSO, setiap solusi kandidat disebut partikel. Partikel-partikel ini bergerak di dalam ruang pencarian dengan dua atribut utama, yaitu posisi dan kecepatan. Setiap partikel menyimpan solusi terbaik yang pernah ditemukannya (*personal best*/pbest) serta mengakses solusi terbaik global yang ditemukan seluruh populasi (*global best*/gbest) (Naiborhu et al., 2013), (Lazinica, 2009).

Perbaruan kecepatan dan posisi partikel dilakukan menggunakan persamaan berikut.

$$v_i^{k+1} = v_i^k + c_1 r_1 \left(p_i^k - x_i^k\right) + \ c_2 r_2 (p_g^k - x_i^k) \tag{26}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{27}$$

dengan

v_i^k : kecepatan partikel ke-i pada iterasi ke-k,
x_i^k : posisi partikel ke-i pada iterasi ke-k,
p_i^k : posisi terbaik individu (personal best / pbest) partikel ke-i,
p_g^k : posisi terbaik global (global best / gbest) dari seluruh partikel,
c_1, c_2 : koefisien akselerasi (menentukan bobot pengaruh pbest dan gbest),
$r_1, r_2 \in [0,1]$: bilangan acak *uniform.*

Untuk meningkatkan performa, PSO sering dilengkapi dengan bobot inersia (w) sehingga persamaan kecepatan menjadi (Soleimani et al., 2009):

$$v_i^{k+1} = w \cdot v_i^k + c_1 r_1 \left(p_i^k - x_i^k\right) + \ c_2 r_2 (p_g^k - x_i^k) \tag{28}$$

dengan w adalah bobot inersia, digunakan untuk mengontrol keseimbangan antara eksplorasi (pencarian global) dan eksploitasi (pencarian lokal). Secara umum, langkah-langkah PSO adalah sebagai berikut.

1) Inisialisasi populasi partikel (posisi dan kecepatan).
2) Evaluasi fungsi objektif setiap partikel.
3) Perbarui nilai pbest dan gbest.
4) Perbarui kecepatan dan posisi partikel dengan Persamaan (26)—(28).
5) Ulangi hingga mencapai kriteria penghentian.

a. Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) merupakan algoritma metaheuristik yang terinspirasi dari perilaku koloni semut dalam menemukan jalur terpendek dari sarang menuju sumber makanan. Secara biologis, semut berkomunikasi menggunakan jejak feromon yang ditinggalkan pada lintasan. Jalur dengan intensitas feromon yang lebih tinggi akan lebih mungkin dipilih oleh semut lain sehingga secara kolektif koloni dapat menemukan solusi optimal. Kemudian, prinsip ini diadaptasi ke dalam algoritma komputasi untuk menyelesaikan permasalahan optimasi dengan ruang solusi yang luas (Dorigo & Stützle, 2010).

Dalam konteks optimasi portofolio, ACO digunakan untuk menentukan bobot alokasi aset yang optimal dengan mempertimbangkan keseimbangan antara tingkat pengembalian (*return*) dan risiko (*risk*). Pada algoritma ini setiap semut merepresentasikan suatu kandidat solusi berupa kombinasi bobot portofolio. Proses eksplorasi solusi dipandu oleh probabilitas pemilihan jalur yang bergantung pada dua komponen utama, yaitu intensitas feromon (τ) dan informasi heuristik (η).

Probabilitas semut k untuk memilih elemen j dari posisi i pada iterasi ke-t didefinisikan sebagai berikut.

$$p_{ij}^k(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum_{l \in J_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}} \tag{29}$$

dengan

$p_{ij}^k(t)$: probabilitas semut k berpindah dari node i ke node j pada iterasi ke-t,
$\tau_{ij}(t)$: intensitas feromon pada jalur i ke j pada iterasi ke-t,
η_{ij} : informasi heuristik yang umumnya didefinisikan sebagai fungsi dari kualitas solusi,
α : parameter yang mengontrol pengaruh relatif dari intensitas feromon,
β : parameter yang mengontrol pengaruh relatif dari informasi heuristik,
J_i^k : himpunan kandidat langkah yang dapat dipilih oleh semut k dari node i.

Melalui mekanisme ini, jalur dengan intensitas feromon tinggi dan informasi heuristik yang menguntungkan akan lebih sering dipilih, sedangkan jalur yang kurang menjanjikan akan

kehilangan daya tarik akibat evaporasi feromon. Secara bertahap, koloni semut akan mengarahkan pencarian menuju solusi portofolio yang mendekati optimal.

Firefly Algorithm (FA)

Firefly Algorithm (FA) diperkenalkan oleh Xin-She Yang pada tahun 2008 sebagai algoritma metaheuristik berbasis populasi yang terinspirasi dari perilaku kunang-kunang dalam menarik satu sama lain melalui cahaya (Yang & He, 2013). Dalam konteks optimasi, setiap kunang-kunang diasosiasikan dengan solusi, dan intensitas cahaya mencerminkan kualitas solusi (*fitness*). Semakin terang (lebih besar nilai fungsi objektif), semakin menarik kunang-kunang lain untuk mendekat. Daya tarik antara dua kunang-kunang *i* dan *j* dengan jarak r_{ij} dirumuskan sebagai berikut.

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \tag{30}$$

dengan

β_0 : daya tarik maksimum (saat r_{ij} = 0),
γ : koefisien penyerapan cahaya,
r_{ij} : jarak Euclidean antara posisi kunang-kunang *i* dan *j*.

Pergerakan kunang-kunang i menuju j (jika j lebih terang) pada iterasi t dituliskan sebagai berikut.

$$x_i^{t+i} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha\varepsilon_i^t \tag{31}$$

dengan

x_i^t : posisi kunang-kunang *i* pada iterasi *t*,
α : langkah acak (faktor randomisasi),
ε_i^t : vektor acak.

Artificial Bee Colony (ABC)

Artificial Bee Colony (ABC) merupakan algoritma metaheuristik berbasis *swarm intelligence* yang meniru perilaku koloni lebah dalam mencari sumber makanan (Furqan et al., 2022). Algoritma ini terdiri dari tiga jenis lebah, yaitu *employed bee, onlooker bee,* dan *scout bee* (Forcael et al., 2024). Proses ini dimulai dengan inisialisasi populasi dengan sejumlah solusi awal dibangkitkan secara acak dalam ruang pencarian. Posisi sumber makanan ke-*i* dirumuskan sebagai berikut.

$$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \tag{32}$$

dengan

x_{ij} : populasi solusi ke-*i* pada dimensi ke-*j*,
x_j^{min} : batas bawah variabel ke-*j*,
x_j^{max} : batas atas variabel ke-*j*,
$rand(0,1)$: bilangan acak antara 0 dan 1.

Setelah inisialisasi, *employed bee* mengeksplorasi solusi baru di sekitar sumber maanan yang ada menggunakan persamaan berikut.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{33}$$

dengan

v_{ij} : kandidat solusi baru,
x_{ij} : posisi awal,
x_{kj} : posisi solusi lain yang dipilih secara acak ($k \neq i$),
ϕ_{ij} : bilangan acak dalam rentang $[-1,1]$.

Selanjutnya, *onlooker bee* memilih sumber makanan berdasarkan probabilitas yang proporsional dengan nilai *fitness* masing-masing solusi. Probabilitas tersebut dihitung menggunakan persamaan berikut.

$$p_i = \frac{fit_i}{\sum_{j=1}^{N} fit_j} \tag{34}$$

dengan

p_i : probabilitas pemilihan solusi ke-*i*,
fit_i : nilai *fitness* solusi tersebut,
N : jumlah total solusi dalam populasi.

Semakin tinggi nilai *fitness,* semakin besar solusi tersebut dipilih. Jika suatu solusi tidak mengalami perbaikan dalam sejumlah siklus tertentu (limit), solusi tersebut ditinggalkan, dan *scout bee* akan membangkitkan solusi baru secara acak menggunakan persamaan inisialisasi

awal (Forcael et al., 2024). Proses ini diulang terus hingga mencapai jumlah iterasi maksimum atau konvergensi tercapai. Pada akhir proses solusi terbaik yang diperoleh dianggap sebagai hasil optimal.

Cuckoo Search (CS)

Algoritma Cuckoo Search merupakan algoritma metaheuristik yang dikembangkan oleh Yang dan Deb pada tahun 2009. Algoritma ini terinspirasi dari perilaku burung cuckoo dalam berkembang biak. Burung cuckoo tidak mengerami telurnya melainkan mencari sarang burung lain lalu meletakkan telur pada sarang tersebut agar dierami oleh burung inang (Nugroho et al., 2023). Dalam algoritma ini, setiap solusi dianalogikan sebagai telur cuckoo yang diletakkan dalam sebuah sarang. Jika telur tersebut memiliki kualitas lebih baik dibandingkan telur asli maka sarang akan mempertahankannya. Sebaliknya, dengan probabilitas tertentu, telur asing dapat dikenali dan dibuang sehingga sarang digantikan dengan solusi baru (Xiong et al., 2023).

Proses pencarian solusi baru dilakukan menggunakan Levy Flight, yaitu pencarian langkah yang terinspirasi dari gerak lalat buah saat mencari makanan dengan mekanisme langkah acak dengan distribusi *heavy-tailed*. Persamaan pergerakan solusi adalah sebagai berikut.

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Levy(\lambda) \tag{35}$$

dengan

$x_i^{(t)}$: posisi solusi ke-*i* pada iterasi ke-*t*,
$\alpha > 0$: skala langkah (*step size*) yang biasanya bernilai kecil,
$\oplus$: perkalian elemen-per-elemen,
$Levy(\lambda)$: distribusi Levy dengan parameter λ (biasanya $1 < \lambda \leq 3$).

Distribusi Levy memiliki fungsi densitas peluang sebagai berikut.

$$Levy \sim u = t^{-\lambda}, \quad (1 < \lambda \leq 3) \tag{36}$$

Selain itu, proses pembaruan sarang dilakukan dengan mekanisme probabilistik. Dengan probabilitas p_a, beberapa sarang digantikan oleh solusi baru untuk menjaga keberagaman populasi (Xiong et al., 2023). Proses ini dituliskan pada persamaan berikut.

$$x_i^{(t+1)} = x_{min} + rand(0,1)(x_{max} - x_{min}) \tag{37}$$

dengan

x_{min} : batas bawah ruang pencarian,
x_{max} : batas atas ruang pencarian,
$rand(0,1)$: bilangan acak uniform pada interval [0,1].

Proses iterasi Cuckoo Search berlangsung hingga jumlah maksimum iterasi tercapai atau kondisi konvergensi terpenuhi. Solusi terbaik yang diperoleh dari seluruh populasi dianggap sebagai solusi optimal.

Genetics Algorithm (GA)

Metode Genetics Algorithm pertama kali diperkenalkan oleh John Holland pada tahun 1960 dan kemudian dipopulerkan oleh David Goldberg pada tahun 1980. Genetics Algithm merupakan metode metaheuristik yang berlandaskan pada prinsip genetika dan proses seleksi alam dalam teori evolusi Darwin (Fahria & Kurtiawan, 2022). Pendekatan ini meniru mekanisme evolusi biologis sehingga istilah-istilah seperti seleksi, *crossover*, dan mutasi digunakan untuk menggambarkan proses pencarian optimal (Azim et al., 2021).

Proses optimasi portofolio dengan Genetics Algorithm diawali dengan pembentukan kromosom yang merepresentasikan solusi dari permasalahan. Setiap kromosom disusun menggunakan metode *real-coded* yang berisi gen-gen berupa bobot saham dalam portofolio (Wahyuni et al., 2017). Pembentukan populasi awal dilakukan dengan menghasilkan sejumlah kromosom secara acak berdasarkan parameter kontrol yang telah ditentukan. Selanjutnya, dilakukan evaluasi fitness untuk menilai kualitas setiap kromosom. Nilai *fitness* menunjukkan seberapa baik suatu portofolio dalam menghasilkan *return* dengan risiko tertentu. Perhitungan *fitness* menggunakan persamaan berikut.

$$fitness = \frac{\bar{R}_p}{\sigma_p} = \frac{\alpha_p + \beta_p \bar{R}_m}{\sqrt{\beta_p^2 \sigma_m^2 + \left(\sum_{i=1}^{n} w_i \sigma_{\epsilon_i}\right)^2}} \tag{38}$$

dengan

$\bar{R}_p$: *expected return* portofolio,

σ_p : risiko portofolio.

Tahap berikutnya adalah seleksi kromosom, yaitu memilih kromosom terbaik berdasarkan nilai *fitness* tertinggi agar dapat digunakan sebagai populasi pada generasi berikutnya (Lestari et al., 2019). Proses seleksi menerapkan metode *elitism* dengan mempertahankan kromosom terbaik. Proses *crossover* dilakukan untuk menghasilkan *offspring* dari dua kromosom induk yang terpilih secara acak. *Offspring* yang dihasilkan dapat menggunakan persamaan berikut.

$$C_1 = P_1 + \alpha(P_2 - P_1) \quad (39)$$

$$C_2 = P_2 + \alpha(P_1 - P_2) \quad (40)$$

dengan

C_1, C_2 : *offspring* yang dihasilkan dari proses *crossover,*

P_1, P_2 : induk yang terpilih berdasarkan bilangan acak yang dibangkitkan,

α : nilai yang dibangkitkan pada interval yang telah ditentukan.

Tahap mutasi dilakukan untuk menjaga keragaman populasi dengan mengubah nilai satu atau beberapa gen pada kromosom. Setelah proses mutasi, kromosom yang tidak memenuhi kendala diperbaiki menggunakan persamaan berikut.

$$New\ gen_i = \frac{isi\ gen_i}{\sum_{i=1}^{n} isi\ gen_i} \quad (41)$$

dengan

$isi\ gen$: nilai yang terdapat dalam gen,

n : jumlah gen dalam satu kromosom.

Tahapan-tahapan tersebut diulang hingga kondisi berhenti tercapai, yaitu ketika jumlah iterasi maksimum terpenuhi atau tidak ada peningkatan signifikan pada nilai *fitness*.

Jellyfish Search Optimization (JSO)

Jellyfish Search Optimization (JSO) merupakan algoritma *metaheuristik* berbasis *swarm intelligence* yang terinspirasi dari perilaku ubur-ubur dalam mencari makanan di lautan (Chou & Molla, 2022). Algoritma ini memiliki dua mekanisme utama, yaitu eksplorasi dengan mengikuti arus laut dan eksploitasi dengan bergerak di dalam kelompok. Pergantian kedua mekanisme diatur oleh fungsi kontrol waktu sehingga keseimbangan pencarian solusi tetap terjaga. Proses JSO diawali dengan inisialisasi populasi ubur-ubur yang dibangkitkan secara acak menggunakan *logistic chaotic map* dengan persamaan sebagai berikut (Chou & Molla, 2022).

$$X_{i+1} = \eta X_i(1 - X_i), \quad 0 \le X_i \le 1 \quad (42)$$

dengan

X_i : posisi ubur-ubur ke-*i,*

η : parameter *chaos.*

Pada tahap eksplorasi ubur-ubur mengikuti arus laut karena daerah tersebut kaya akan nutrisi. Arus laut dihitung menggunakan persamaan berikut.

$$\overrightarrow{trend} = X^* - \beta \times rand(0,1) \times \mu \quad (43)$$

dengan

X^* : posisi terbaik,

μ : rata-rata posisi seluruh ubur-ubur,

β : koefisien distribusi.

Kemudian, posisi baru ubur-ubur dihitung menggunakan persamaan berikut.

$$X_i(t+1) = X_i(t) + rand(0,1) \times \overrightarrow{trend} \quad (44)$$

$$X_i(t+1) = X_i(t) + rand(0,1) \times X^* - \beta \times rand(0,1) \times \mu \quad (45)$$

dengan

$X_i(t)$: posisi ubur-ubur sekarang,

$X_i(t+1)$: posisi baru ubur-ubur,

$rand(0,1)$: bilangan acak antara 0–1,

β : koefisien distribusi.

Sementara itu, pada tahap eksploitasi, ubur-ubur bergerak di dalam *swarm* dengan dua tipe pergerakan. Pergerakan tipe A (pasif) dilakukan dengan berpindah di sekitar posisi awal menggunakan persamaan berikut.

$$X_i(t+1) = X_i(t) + \gamma \times rand(0,1) \times (Ub - Lb) \quad (46)$$

dengan

γ : faktor skala,

Ub : batas atas ruang pencarian,

Lb : batas bawah ruang pencarian.

Pergerakan tipe B (aktif) dilakukan dengan mendekati atau menjauhi individu lain bergantung pada nilai fungsi objektif $f(X)$. Arah pergerakan ini dirumuskan sebagai berikut.

$$Direction = \begin{cases} X_j(t) - X_i(t), & f(X_i) \geq f(X_j) \\ X_i(t) - X_j(t), & f(X_i) < f(X_i) \end{cases} \quad (47)$$

Kemudian, posisi baru dihitung menggunakan persamaan berikut.

$$X_i(t+1) = X_i(t) + rand(0,1) \times Direction \quad (48)$$

Peralihan antara tahap eksplorasi dan eksploutasi dikendalikan oleh fungsi kontrol waktu yang dirumuskan sebagai berikut.

$$c(t) = \left| \left(1 - \frac{t}{MaxIter}\right) \times (2 \times rand(0,1) - 1) \right| \quad (49)$$

Jika $c(t) \geq 0{,}5$, ubur-ubur mengikuti arus laut (eksplorasi). Sementara itu, jika $c(t) < 0{,}5$, ubur-ubur bergerak di dalam *swarm* (eksploitasi). Setiap iterasi, dilakukan pengecekan batas (*boundary condition*) untuk memastikan posisi ubur-ubur tetap berada dalam ruang pencarian (Chou & Molla, 2022). Proses ini dilanjutkan hingga mencapai jumlah iterasi maksimum atau konvergensi.

Sharpe Ratio

Analisis ini dilakukan untuk mengukur dan mengevaluasi kinerja portofolio investasi dengan memperhitungkan risiko melalui indikator *sharpe ratio*. Metode ini dipilih karena *sharpe ratio* merupakan salah satu alat ukur *risk-adjusted return* yang paling banyak digunakan dalam kajian keuangan dan investasi (Kan et al., 2024). *Sharpe* r*atio* memberikan ukuran yang mengintegrasikan imbal hasil portofolio relatif terhadap tingkat pengembalian bebas risiko dan volatilitas *return* sehingga mampu merefleksikan efektivitas manajemen risiko investasi (Kourtis, 2016).

Perhitungan *sharpe* r*atio* dilakukan dengan menggunakan formula sebagai berikut (Bajaj Finserv, 2025).

$$S_a = \frac{E[R_a - R_b]}{\sigma_a} \quad (50)$$

dengan

R_a : rata-rata *return* portofolio,
R_b : *return* bebas risiko (*risk-free rate*),
σ_a : standar deviasi return portofolio sebagai ukuran risiko volatilitas investasi.

Klasifikasi nilai *sharpe ratio* beserta tingkat risiko dan penilaian performanya adalah sebagai berikut.

Tabel 1. Klasifikasi *Sharpe Ratio*

Rentang *Sharpe Ratio*	Tingkat Risiko	Kategori Kinerja
< 1,00	Rendah	Buruk
1,00 – 1,99	Sedang	Baik
2,00 – 2,99	Tinggi	Sangat baik
> 3,00	Sangat Tinggi	Unggul

Kemampuan *sharpe ratio* dalam mencerminkan kinerja *real return* disesuaikan dengan risiko yang diambil menjadikannya indikator penting dalam manajemen investasi.

Value at Risk (VaR)

Value at Risk (VaR) merupakan salah satu ukuran risiko yang digunakan untuk mengestimasi kerugian maksimum yang mungkin terjadi pada suatu portofolio dalam periode tertentu dengan tingkat kepercayaan (*confidence level*) tertentu (Jorion, 2011). VaR memberikan estimasi mengenai besarnya kerugian yang mungkin terjadi akibat fluktuasi pasar dengan probabilitas yang telah ditentukan. Secara umum, metode perhitungan VaR dapat diklasifikasikan ke dalam tiga pendekatan utama, yaitu sebagai berikut.

a. Metode Parametrik (*Variance-Covariance Method*)

Pendekatan ini mengasumsikan bahwa return aset mengikuti distribusi normal. Formula dasar yang digunakan adalah sebagai berikut.

$$VaR = Z_\alpha \cdot \sigma_p \cdot \sqrt{t} \quad (51)$$

dengan

Z_α : nilai kritis distribusi normal sesuai tingkat kepercayaan,

σ_p : standar deviasi portofolio,
$\sqrt{t}$: horizon waktu pengukuran.

b. Metode Simulasi Historis (*Historical Simulation*)
Pendekatan ini menggunakan data historis *return* tanpa mengasumsikan distribusi tertentu. Nilai VaR diperoleh dengan mengurutkan data *return* dari yang terburuk hingga terbaik. Kemudian, mengambil nilai pada kuantil sesuai tingkat kepercayaan.

c. Metode Simulasi Monte Carlo (Monte Carlo *Simulation*)
Pada pendekatan ini *return* aset disimulasikan secara acak dalam jumlah besar menggunakan asumsi distribusi probabilitas tertentu. Dari hasil simulasi, nilai VaR diperoleh berdasarkan kuantil distribusi *return* simulasi.

Selain berdasarkan metode perhitungan, VaR juga dapat dikembangkan ke dalam beberapa turunan analisis, antara lain Individual VaR (risiko pada aset tunggal), portofolio VaR (risiko gabungan seluruh aset dalam portofolio), serta *Incremental* VaR dan *Marginal* VaR yang mengukur kontribusi dan sensitivitas suatu aset terhadap total risiko portofolio.

Simulasi *Latent Performance Investment*

Pendekatan *latent performance investment* pada penelitian ini dikembangkan sebagai kerangka analisis untuk mengevaluasi kinerja portofolio dalam jangka panjang dengan menggabungkan elemen *unsupervised learning* dan *metaheuristik optimization*. Tahapan awal melibatkan proses klasterisasi saham menggunakan data historis periode Januari 2015 hingga Desember 2019. Proses klasterisasi ini bertujuan mengidentifikasi kelompok saham dengan perilaku dan karakteristik performa yang homogen, seperti volatilitas, rata-rata pengembalian, serta korelasi antar aset. Dengan demikian, setiap *cluster* diharapkan mewakili suatu profil risiko-*return* tertentu yang dapat menjadi dasar pembentukan portofolio yang lebih stabil.

Tahap berikutnya melibatkan optimisasi portofolio menggunakan beberapa algoritma berbasis metaheuristik yang dirancang untuk menyeimbangkan antara tingkat pengembalian (*return*) dan risiko (*risk*). Proses optimisasi ini juga memungkinkan penerapan berbagai kendala, seperti *skewness*, kurtosis, dan Value at Risk (Ren et al., 2018). Rumus perhitungannya adalah sebagai berikut.

$$CAGR = \left(\frac{V_f}{V_i}\right)^{\frac{1}{n}} - 1 \tag{52}$$

dengan
V_f : nilai akhir
V_i : nilai awal investasi,
n : jumlah tahun investasi berlangsung.

CAGR digunakan untuk mengevaluasi efisiensi strategi investasi jangka panjang yang dihasilkan oleh proses optimisasi. Nilai CAGR yang lebih tinggi menunjukkan portofolio memiliki tingkat pertumbuhan yang lebih baik dengan asumsi pertumbuhan tahunan yang konsisten.

C. Hasil dan Pembahasan

Pemilihan Saham untuk Portofolio

Pemilihan saham merupakan tahap penting dalam pembentukan portofolio karena menentukan ruang lingkup analisis dan hasil optimasi. Penelitian ini menggunakan 25 saham dari lma sektor utama di Bursa Efek Indonesia (BEI), yaitu *Consumer Non-Cyclicals*, *Financial*, *Energy* & *Mining*, *Infrastructure* & *Telecom*, dan *Healthcare* dengan masing-masing sektor diwakili oleh lima saham. Pemilihan dilakukan secara representatif lintas sektor dengan mempertimbangkan kapitalisasi pasar, likuiditas, stabilitas kinerja historis, serta ketersediaan data harga. Untuk menjaga objektivitas, saham dipilih secara acak (*random sampling*) dari kelompok berkapitalisasi menengah hingga besar yang aktif diperdagangkan di BEI.

Pratinjau Data Saham yang Digunakan

Adapun rincian saham yang digunakan beserta alasan pemilihannya adalah sebagai berikut.

a. *Consumer Non-Cyclicals* (Konsumsi Primer)

Sektor konsumsi primer dipilih karena bersifat defensif dan relatif stabil terhadap fluktuasi ekonomi makro. Saham yang digunakan adalah sebagai berikut.

- ICBP.JK (Indofood CBP Sukses Makmur Tbk) karena memiliki kinerja stabil dan merupakan produsen makanan olahan terbesar di Indonesia.

- INDF.JK (Indofood Sukses Makmur Tbk) karena induk usaha besar di sektor pangan dengan diversifikasi produk luas.
- MYOR.JK (Mayora Indah Tbk) karena dikenal dengan pertumbuhan ekspor yang konsisten.
- AMRT.JK (Sumber Alfaria Trijaya Tbk/Alfamart) karena merepresentasikan sektor ritel modern dengan tingkat likuiditas tinggi.
- UNVR.JK (Unilever Indonesia Tbk) karena perusahaan multinasional dengan fundamental keuangan yang kuat dan dividen stabil.

b. *Energy & Mining* (Energi dan Pertambangan)

Sektor ini dipilih karena memiliki potensi imbal hasil tinggi dan berkaitan erat dengan harga komoditas global. Saham yang digunakan adalah sebagai berikut.

- ADRO.JK (Adaro Energy Indonesia Tbk) karena produsen batu bara terbesar kedua di Indonesia.
- PTBA.JK (Bukit Asam Tbk) karena perusahaan tambang batu bara milik negara dengan kinerja keuangan kuat.
- ANTM.JK (Aneka Tambang Tbk) karena merepresentasikan subsektor logam mulia dan nikel.
- INCO.JK (Vale Indonesia Tbk) karena berfokus pada produksi nikel untuk industri baterai kendaraan listrik.
- ITMG.JK (Indo Tambangraya Megah Tbk) karena perusahaan batu bara dengan tingkat efisiensi produksi tinggi.

c. *Financial* (Keuangan)

Sektor keuangan dipilih karena memiliki kontribusi besar terhadap indeks pasar dan menggambarkan stabilitas sistem keuangan nasional. Saham yang digunakan adalah sebagai berikut.

- BBRI.JK (Bank Rakyat Indonesia Tbk) karena memiliki jangkauan nasabah terbesar di Indonesia.
- BBNI.JK (Bank Negara Indonesia Tbk) karena mewakili sektor perbankan korporasi dan pembiayaan ekspor.
- BMRI.JK (Bank Mandiri Tbk) karena memiliki total aset terbesar di Indonesia.
- BBCA.JK (Bank Central Asia Tbk) karena dikenal dengan efisiensi tinggi dan profitabilitas stabil.
- BNGA.JK (CIMB Niaga Tbk) karena memiliki aktivitas perdagangan saham yang cukup likuid.

d. *Healthcare* (Kesehatan)

Sektor kesehatan dipilih karena sifatnya yang defensif dan pertumbuhannya relatif stabil pascapandemi. Saham yang digunakan adalah sebagai berikut.

- KLBF.JK (Kalbe Farma Tbk) karena merupakan produsen farmasi terbesar di Indonesia.
- MERK.JK (Merck Tbk) karena perusahan perusahaan farmasi multinasional dengan fundamental kuat.
- SILO.JK (Siloam International Hospitals Tbk) karena memiliki jaringan rumah sakit swasta terbesar di Indonesia.
- SIDO.JK (Sido Muncul Tbk) karena merupakan perusahaan jamu dan farmasi dengan performa keuangan konsisten.
- DVLA.JK (Darya-Varia Laboratoria Tbk) karena berfokus pada produksi dan distribusi produk kesehatan serta farmasi premium yang memiliki prospek pertumbuhan stabil di sektor kesehatan nasional.

e. *Infrastructure & Telecom* (Infrastruktur dan Telekomunikasi)

Sektor ini memiliki peran strategis dalam pembangunan nasional dan perkembangan teknologi digital. Saham yang digunakan adalah sebagai berikut.

- TLKM.JK (Telkom Indonesia Tbk) karena merupakan BUMN terbesar di bidang telekomunikasi dan memiliki peran penting dalam pengembangan infrastruktur digital Indonesia.
- ISAT.JK (Indosat Ooredoo Hutchison Tbk) karena merupakan perusahaan telekomunikasi terkemuka dengan basis pelanggan luas dan kontribusi signifikan terhadap layanan komunikasi nasional..
- EXCL.JK (XL Axiata Tbk) karena memiliki strategi ekspansi jaringan yang agresif serta kinerja pertumbuhan pelanggan yang stabil.
- JSMR.JK (Jasa Marga Tbk) karena merepresentasikan subsektor infrastruktur jalan tol dan memiliki peran penting dalam mendukung mobilitas serta konektivitas nasional.
- PGAS.JK (Perusahaan Gas Negara Tbk) karena perusahaan energi yang berperan dalam distribusi gas bumi nasional.

Data harga saham yang digunakan dalam analisis ini diperoleh dari Yahoo Finance dengan periode pengamatan mulai 1 Januari 2015 hingga 31 Desember 2019. Rentang waktu lima tahun tersebut dipilih karena mencerminkan kondisi perekonomian nasional yang relatif stabil tanpa gangguan besar seperti pandemi COVID-19 atau krisis global. Seluruh saham juga aktif diperdagangkan selama periode tersebut sehingga data historis yang diperoleh lengkap dan dapat diandalkan. Periode ini dianggap ideal untuk menghasilkan analisis yang konsisten dalam perhitungan *expected return* dan *volatilitas* sebagai dasar proses optimasi portofolio. Dengan pertimbangan tersebut, pemilihan periode 2015–2019 diharapkan mampu mendukung pembentukan portofolio yang representatif, terdiversifikasi optimal, serta menghasilkan analisis yang valid secara empiris dan metodologis.

Analisis Dasar Data Saham

Tahap awal penelitian adalah menghitung parameter dasar dari masing-masing saham, yaitu rata-rata tingkat pengembalian (*mean return*) dan volatilitas (*standard deviation*). Kedua ukuran ini merupakan indikator fundamental dalam analisis portofolio karena menggambarkan potensi keuntungan yang diharapkan dan besarnya fluktuasi harga yang harus dihadapi *investor*.

Mean return diperoleh dari rata-rata *log-return* harian saham selama periode pengamatan (2015–2019). Nilai ini mencerminkan estimasi tingkat keuntungan rata-rata yang dapat dihasilkan suatu saham. Semakin tinggi nilai *mean return*, semakin besar potensi keuntungan yang ditawarkan. Sementara itu, *standard deviation* digunakan untuk mengukur volatilitas atau risiko saham. Nilai yang lebih tinggi menunjukkan pergerakan harga yang lebih fluktuatif, sehingga *investor* menghadapi ketidakpastian lebih besar.

Tabel 2. *Mean Return* dan *Standard Deviation* Saham (2015–2019)

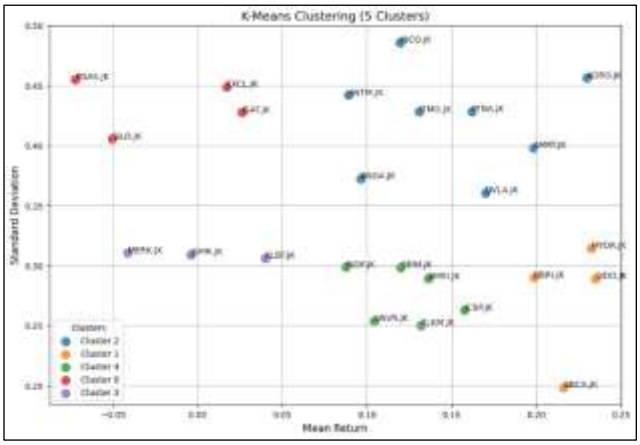
Sektor	Saham	*Mean Return*	*Standard Deviation*
Consumer Non-Cyclicals	AMRT.JK	0,198	0,398
	ICBP.JK	0,158	0,263
	INDF.JK	0,087	0,299
	MYOR.JK	0,233	0,314
	UNVR.JK	0,104	0,254
Financial	BBCA.JK	0,216	0,198
	BBNI.JK	0,120	0,299
	BBRI.JK	0,199	0,290
	BMRI.JK	0,136	0,289
	BNGA.JK	0,096	0,372
Energy & Mining	ADRO.JK	0,230	0,456
	ANTM.JK	0,089	0,442
	INCO.JK	0,120	0,486
	ITMG.JK	0,131	0,428
	PTBA.JK	0,162	0,428
Infrastructure & Telecom	EXCL.JK	0,017	0,449
	ISAT.JK	0,026	0,428
	JSMR.JK	-0,004	0,309
	PGAS.JK	-0,072	0,455
	TLKM.JK	0,132	0,250
Healthcare	DVLA.JK	0,170	0,361
	KLBF.JK	0,040	0,306
	MERK.JK	-0,041	0,311
	SIDO.JK	0,235	0,289
	SILO.JK	-0,051	0,406

Pengelompokkan Data Saham Menggunakan Analisis Clustering

Karakteristik *mean return* dan *risk* (standar deviasi) diterapkan untuk memudahkan analisis dalam pemilihan aset yang akan membentuk portofolio investasi. Berdasarkan dua indikator tersebut, dilakukan pengelompokan (*clustering*) saham menggunakan tiga metode, yaitu K-Means *Clustering*, Fuzzy C-Means *Clustering*, dan Agglomerative *Clustering*. Proses ini bertujuan untuk mengelompokkan saham ke dalam klaster yang memiliki tingkat risiko dan imbal hasil yang serupa sehingga pemilihan aset dapat dilakukan secara lebih sistematis. Evaluasi hasil klaster dilakukan menggunakan *silhouette score* dan hasil perbandingannya disajikan pada Tabel 3.

Tabel 3. Perbandingan *Silhouette Score* Setiap Metode

Metode *Clustering*	Jumlah Klaster	*Silhouette Score*
K-Means *Clustering*	**5**	**0,433**
Fuzzy C-Means *Clustering*	5	0,432
Agglomerative *Clustering*	5	0,394



Gambar 2. Visualisasi Hasil K-Means *Clustering*

Berdasarkan Tabel 3, metode K-Means *Clustering* memiliki nilai *Silhouette Score* tertinggi, yaitu 0,4333 sehingga dipilih sebagai metode yang paling optimal dalam pengelompokan data saham pada analisis ini. Hasil pengelompokan dengan K-Means menghasilkan lima klaster dengan karakteristik sebagai berikut.

- **Klaster 0 (merah)**, yaitu saham dengan volatilitas tertinggi dan *mean return* rendah, menunjukkan saham yang berisiko tinggi dengan potensi kerugian.
- **Klaster 1 (oranye)**, yaitu saham dengan volatilitas rendah hingga sedang dan *mean return* tinggi sehingga menjadi kelompok paling ideal untuk portofolio investasi.
- **Klaster 2 (biru)**, yaitu saham dengan volatilitas relative tinggi dan *mean return* sedang hingga tinggi, mencerminkan saham-saham dengan potensi keuntungan menarik tetapi disertai volatilitas yang lebih besar.
- **Klaster 3 (ungu)**, yaitu saham dengan volatilitas moderat dan *mean return* rendah hingga sedang, menunjukkan performa yang cenderung stabil tetapi dengan potensi imbal hasil yang tidak terlalu besar.
- **Klaster 4 (hijau)**, yaitu saham dengan volatilitas moderat hingga rendah dan *mean return* sedang, menggambarkan saham-saham berprofil konservatif dengan kestabilan harga yang relatif baik.

Pada tahap selanjutnya Klaster 1 dipilih sebagai kelompok saham yang akan dianalisis lebih lanjut karena memiliki kombinasi *return* yang tinggi dengan risiko yang relatif terkendali. Pemilihan ini selaras dengan prinsip *risk-return tradeoff*, yaitu investor cenderung memilih aset yang mampu memberikan imbal hasil optimal pada tingkat risiko tertentu (Bannier et al., 2023).

Terdapat empat saham yang termasuk dalam Klaster 1, yaitu BBRI.JK, MYOR.JK, SIDO.JK, dan BBCA.JK. Saham-saham tersebut akan menjadi kandidat dalam tahap optimisasi portofolio yang dilakukan dengan memaksimalkan *sharpe ratio* menggunakan delapan algoritma metaheuristik.

Pembobotan Portofolio Metaheuristik

Evaluasi Kinerja Pembobotan Portofolio Metaheuristik

Tahap ini bertujuan untuk mengevaluasi kinerja delapan algoritma metaheuristik dalam menentukan bobot optimal portofolio saham, yaitu Grey Wolf Optimization (GWO), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Firefly Algorithm (FA), Artificial Bee Colony (ABC), Cuckoo Search (CS), Genetic Algorithm (GA), dan Jellyfish Search Optimization (JSO).

Evaluasi dilakukan berdasarkan tiga indikator utama, yaitu *sharpe ratio*, *historical* Value at Risk (VaR), dan *probability of loss* (%). Hasil perbandingan kinerja masing-masing metode metaheuristik ditampilkan pada Tabel 4.

Tabel 4. Evaluasi Kinerja Metode Metaheuristik

Metode	*Sharpe Ratio*	*Historical* VaR	*Prob of Loss* (%)
GWO	**1,392268**	**0,015390**	**42,44**

Metode	*Sharpe Ratio*	*Historical* VaR	*Prob of Loss* (%)
PSO	1,391407	0,015290	42,28
ACO	1,391904	0,015498	42,60
FA	1,392268	0,015390	42,60
ABC	1,392268	0,015391	42,60
CS	0,213298	0,027528	48,43
GA	1,162710	0,022471	43,23
JSO	1,392233	0,015460	42,60

Hasil evaluasi menunjukkan bahwa sebagian besar metode metaheuristik, yaitu GWO, PSO, ACO, FA, ABC, dan JSO, menghasilkan kinerja yang relatif serupa pada seluruh indikator evaluasi. Kemiripan nilai Sharpe ratio, historical VaR, dan *probability of loss* mengindikasikan bahwa metode-metode tersebut memiliki karakteristik dan kemampuan optimisasi yang hampir sebanding dalam menyelesaikan permasalahan optimisasi portofolio. Meskipun menggunakan mekanisme pencarian yang berbeda, seluruh algoritma tersebut sama-sama menekankan keseimbangan antara eksplorasi dan eksploitasi, sehingga cenderung berkonvergensi pada solusi portofolio yang berada pada wilayah optimum yang sama atau sangat berdekatan.

Di antara metode-metaheuristik tersebut, metode Grey Wolf Optimization (GWO) menunjukkan performa terbaik dengan *sharpe ratio* tertinggi sebesar 1,392268, disertai nilai *historical* VaR yang rendah (0,015390) dan *probability of loss* sebesar 42,44% yang juga relatif kecil dibandingkan metode lainnya. Hal ini menunjukkan bahwa GWO mampu menghasilkan kombinasi portofolio yang memberikan imbal hasil paling efisien dengan tingkat risiko terukur dan potensi kerugian yang minimal. Oleh karena itu, metode GWO dipilih sebagai algoritma optimisasi utama dalam tahap pembobotan portofolio dan pada tahap simulasi *Latent Performance Invesment*.

Pembobotan Portofolio Metaheuristik Menggunakan Metode Terbaik

Setelah metode *Grey Wolf Optimizer* (GWO) ditetapkan sebagai algoritma terbaik, langkah selanjutnya adalah menentukan bobot optimal ternormalisasi (*normalized weight*) masing-masing saham yang termasuk dalam klaster terpilih (BBCA.JK, BBRI.JK, MYOR.JK, dan SIDO.JK).

Tabel 5. Bobot Setiap Saham Metode GWO

Nama Saham	Bobot (*Normalized Weight*)
BBCA.JK	0,525
BBRI.JK	0,023
MYOR.JK	0,203
SIDO.JK	0,249

Hasil perhitungan yang disajikan pada Tabel 5 menunjukkan bahwa saham BBCA.JK memperoleh bobot terbesar, yaitu sebesar 0,524903. Hal tersebut berarti bahwa saham BBCA.JK memberikan kontribusi paling dominan dalam pembentukan portofolio optimal. Saham MYOR.JK dan SIDO.JK memiliki bobot moderat, masing-masing sebesar 0,203522 dan 0,248864, sedangkan BBRI.JK memiliki bobot paling kecil sebesar 0,022712.

Kombinasi bobot ini menghasilkan portofolio dengan profil risiko dan imbal hasil yang seimbang, yaitu dengan alokasi terbesar diberikan kepada saham dengan stabilitas tinggi dan potensi return yang optimal sesuai dengan prinsip efisiensi portofolio (ef*ficient frontier*).

Simulasi Latent Performance Investment

Pembelian Saham Awal

Berdasarkan hasil optimisasi menggunakan algoritma Grey Wolf Optimization (GWO) pada periode Januari 2015 hingga Desember 2019, diperoleh bobot paling optimal untuk untuk empat saham, yaitu BBCA.JK, BBRI.JK, MYOR.JK, dan SIDO.JK. Bobot tersebut merepresentasikan proporsi dana investasi yang dialokasikan ke masing-masing saham.

Simulasi investasi dilakukan pada 30–31 Desember 2019 dengan total modal sebesar Rp500.000.000. Harga penutupan saham pada tanggal tersebut digunakan sebagai acuan untuk menentukan jumlah lembar saham yang dapat dibeli sesuai bobot optimal. Distribusi pembelian saham disajikan pada tabel berikut.

Tabel 6. Jumlah Pembelian Lembar Saham

Saham	Harga Saham pada 30—31 Desember 2019 (Rp)	Lembar Saham
BBCA.JK	5.916,64	44.358

BBRI.JK	2.967,56	3.826
MYOR.JK	1.815,92	56.038
SIDO.JK	482,99	257.629

Nilai realisasi portofolio pada saat pembelian tercatat sebesar Rp499.996.130, mendekati total dana awal yang dialokasikan. Hal ini menunjukkan bahwa proses alokasi modal berbasis bobot optimal dapat diterapkan secara praktis tanpa menimbulkan penyimpangan berarti akibat pembulatan jumlah lembar saham.

Perbandingan Kinerja Portofolio Tahun 2022, 2023, dan 2024

Setelah pembentukan portofolio, dilakukan simulasi strategi *buy-and-hold* tanpa *rebalancing* selama periode pengujian, yaitu dari 30 Desember 2019 hingga 31 Desember 2024. Jumlah lembar saham dari setiap emiten tetap konstan, sedangkan nilai portofolio pada masing-masing tahun dihitung berdasarkan harga penutupan saham pada 31 Desember 2022, 2023, dan 2024.

Tabel 7. Performa Portofolio

Tahun	Nilai Portofolio (Rp)	Return Antarperiode (%)
2019	499.996.130	*baseline*
2022	661.265.408	32,25%
2023	668.032.168	1,02%
2024	732.999.717	9,73%

Dari hasil tersebut, terlihat bahwa nilai portofolio meningkat secara konsisten dari tahun ke tahun dengan total pertumbuhan sekitar 46,6% dalam periode lima tahun. Peningkatan signifikan terjadi antara 2019–2022 yang bertepatan dengan fase pemulihan ekonomi nasional setelah pandemi COVID-19. Pada fase ini saham-saham sektor perbankan dan barang konsumsi mengalami kenaikan harga yang kuat.

Untuk mengukur tingkat pertumbuhan tahunan gabungan, digunakan Compound Annual Growth Rate (CAGR) yang dihitung sebagai berikut.

$$CAGR = \left(\frac{732.999.717}{499.996.130}\right)^{\frac{1}{5}} - 1 = 7{,}95\%$$

Nilai tersebut menunjukkan bahwa portofolio menghasilkan imbal hasil rata-rata sebesar 7,95% per tahun selama periode pengujian tanpa dilakukan perubahan komposisi saham. Dengan kata lain, strategi pasif berbasis bobot optimal GWO tetap memberikan pertumbuhan positif meskipun kondisi pasar mengalami fluktuasi yang cukup besar dalam rentang waktu tersebut.

D. Kesimpulan

Analisis ini mengimplementasikan pendekatan Optimasi Portofolio Saham Berbasis Metaheuristik dengan pemilihan saham melalui tiga analisis *clustering*, yaitu K-Means, Fuzzy C-Means, dan Agglomerative. Sebanyak 25 saham dari lima sektor utama BEI digunakan dalam periode 2015–2019, yang mencerminkan kondisi ekonomi stabil dan data historis yang lengkap.

Hasil analisis menunjukkan bahwa K-Means *Clustering* merupakan metode terbaik dengan *Silhouette Score* tertinggi, yaitu sebesar 0,433 dengan Klaster 1 yang berisi saham BBCA.JK, BBRI.JK, MYOR.JK, dan SIDO.JK dipilih untuk proses optimasi. Dari delapan algoritma metaheuristik yang diuji, Grey Wolf Optimization (GWO) menghasilkan performa terbaik dengan *Sharpe Ratio* tertinggi sebesar 1,392 serta risiko terendah.

Hasil optimasi GWO menghasilkan bobot portofolio optimal, yaitu BBCA sebesar 52,5%, BBRI sebesar 2,3%, MYOR sebesar 20,3%, dan SIDO sebesar 24,9%. Simulasi investasi sebesar Rp500.000.000 pada akhir 2019 menunjukkan peningkatan nilai portofolio menjadi Rp732.999.717 pada akhir 2024, dengan total pertumbuhan 46,6% dan CAGR sebesar 7,9% per tahun.

Secara keseluruhan, kombinasi *clustering* dan algoritma metaheuristik GWO terbukti efektif dalam membentuk portofolio efisien, menghasilkan imbal hasil stabil, dan menjaga risiko pada tingkat yang terkendali.

E. Referensi

Abdurakhman, Sari, Y. W., Husnaqilati, A., & Maruddani, D. A. I. (2025). Manajemen investasi: Teori dan aplikasi. Gadjah Mada University Press.

Azim, M. F., Azizah, A., & Anggraini, D. (2021). Optimasi bobot portofolio menggunakan algoritma genetika. Jurnal Sains Matematika dan Statistika, 7(1), 58–64. http://dx.doi.org/10.24014/jsms.v7i1.12190

Bajaj Finserv. (2025, January 28). Sharpe ratio - Meaning, formula and calculation. https://www.bajajfinserv.in/investments/what-is-sharpe-ratio

Banaie-Dezfouli, M., Nadimi-Shahraki, M. H., & Beheshti, Z. (2021). R-GWO: Representative-based grey wolf optimizer for solving engineering problems. Applied Soft Computing, 106, 107328. https://doi.org/10.1016/j.asoc.2021.107328

Bannier, C. E., Bofinger, Y., & Rock, B. (2023). The risk-return tradeoff: Are sustainable investors compensated adequately? Journal of Asset Management, 24(3), 165–172. https://doi.org/10.1057/s41260-023-00303-6

Bodie, Z., Kane, A., & Marcus, A. J. (2014). Investments (10th ed.). McGraw-Hill Education.

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. Information Sciences, 237, 82–117. https://doi.org/10.1016/j.ins.2013.02.041

Chou, J. S., & Molla, A. (2022). Recent advances in use of bio-inspired jellyfish search algorithm for solving optimization problems. Scientific Reports, 12, 19157. https://doi.org/10.1038/s41598-022-23121-z

Dani, A. T. R., Wahyuningsih, S., & Rizki, N. A. (2019). Penerapan hierarchical clustering metode agglomerative pada data runtun waktu. Jambura Journal of Mathematics, 1(2), 64–78. http://dx.doi.org/10.34312/jjom.v1i2.2354

Dorigo, M., & Stützle, T. (2010). Ant colony optimization: Overview and recent advances. In M. Gendreau & J.-Y. Potvin (Eds.), Handbook of Metaheuristics (pp. 227–263). Springer.

Erwin, K., & Engelbrecht, A. (2023). Meta-heuristics for portfolio optimization. Soft Computing, 27, 19045–19073. https://doi.org/10.1007/s00500-023-08177-x

Fabozzi, F. J., Gupta, F., & Markowitz, H. M. (2017). The legacy of modern portfolio theory. The Journal of Investing, 26(3), 7–22. https://doi.org/10.3905/joi.2002.319510

Fahria, I., & Kurtiawan, E. (2022). Algoritma genetika dalam pengoptimuman portofolio perusahaan emiten. Jurnal MSA. https://doi.org/10.24252/msa.v8i2.16030

Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2017). Grey wolf optimizer: A review of recent variants and applications. Neural Computing and Applications, 30(2), 413–435. https://doi.org/10.1007/s00521-017-3272-5

Forcael, E., Carriel, I., Opazo-Vega, A., Moreno, F., Orozco, F., Romo, R., & Agdas, D. (2024). Artificial bee colony algorithm to optimize the safety distance of workers in construction projects. Mathematics, 12(13), 2087. https://doi.org/10.3390/math12132087

Furqan, M., Nasution, Y. R., & Khairunnisa, K. (2022). Application of artificial bee colony algorithm to optimize the shortest route to distribute clean water pipes. JOMLAI: Journal of Machine Learning and Artificial Intelligence, 1(2), 125–132. https://doi.org/10.55123/jomlai.v1i2.768

Handayani, C. T. N., & Sitokdana, M. N. (2025). Comparison of K-Means++ and agglomerative hierarchical methods in clustering healthcare workers. INOVTEK Polbeng-Seri Informatika, 10(2), 717–728. https://doi.org/10.35314/pcbrs043

Jorion, P. (2011). Value at risk: The new benchmark for managing financial risk (3rd ed.). McGraw-Hill Education.

Kan, R., Wang, X., & Zheng, X. (2024). In-sample and out-of-sample Sharpe ratios of multi-factor asset pricing models. Journal of Financial Economics. http://dx.doi.org/10.2139/ssrn.3454628

Kourtis, A. (2016). Indexing and risk management using Sharpe ratio. The Journal of Portfolio Management, 42(5), 123–134. https://doi.org/10.3905/jpm.2016.42.5.123

Kurniawan, Y. I., Anugrah, P. R., Sugihono, R. M., Abimanyu, F. A., & Afuan, L. (2023). Pengelompokan prioritas negara yang membutuhkan bantuan menggunakan clustering K-Means dengan Elbow dan Silhouette. Jurnal Pendidikan dan Teknologi Indonesia (JPTI), 3(10), 455–463. https://doi.org/10.52436/1.jpti.343

Lazinica, A. (2009). Particle swarm optimization. InTech.

Lestari, E., Sulistianingsih, E., & Imro'ah, N. (2019). Penentuan portofolio saham optimal menggunakan algoritma genetika. BIMASTER: Buletin Ilmiah Matematika, Statistika dan Terapannya, 8(2). https://doi.org/10.26418/bbimst.v8i2.31534

Levine, R. (2011). Finance and growth: Theory and evidence. In A. J. Auerbach, D. Card, & J. M. Quigley (Eds.), Handbook of Economic Growth (Vol. 1, pp. 865–934). Elsevier. https://doi.org/10.1016/S1574-0684(05)01012-9

Long, W., Jiao, J., Liang, X., & Tang, M. (2018). An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. Engineering Applications of Artificial Intelligence, 68, 63–80. https://doi.org/10.1016/j.engappai.2017.10.024

Martin, M., & Nataliani, Y. (2020). Klasterisasi kinerja karyawan menggunakan algoritma fuzzy C-means. Jurnal AITI, 17(2), 118–129. https://doi.org/10.24246/aiti.v17i2.118-129

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. Advances in Engineering Software, 69, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Musharyadi, F. (2017). Tingkat pemahaman mahasiswa terhadap norma-norma agama Islam menggunakan algoritma K-means clustering. Menara Ilmu: Jurnal Penelitian dan Kajian Ilmiah, 11(78). https://doi.org/10.33559/mi.v11i78.576

Naiborhu, J., Firman, & Mu'tamar, K. (2013). Particle swarm optimization in the exact linearization technic for output tracking of non-minimum phase nonlinear systems. Applied Mathematical Sciences, 7(109), 5427–5442. http://dx.doi.org/10.12988/ams.2013.37425

Nugroho, M. A., Wuryanto, E., & Faqih, K. (2023). Optimizing uncapacitated facility location problem with cuckoo search algorithm based on Gauss distribution. SISTEMASI, 12(2), 361–369. https://doi.org/10.32520/stmsi.v12i2.2467

Nurlaeli, & Artati, D. (2020). Analisis kinerja portofolio saham dengan metode Sharpe, Treynor, dan Jensen. Jurnal Ilmiah Mahasiswa Manajemen, Bisnis dan Akuntansi (JIMMBA), 2(6), 972–990. https://doi.org/10.32639/jimmba.v2i6.690

Putri, G. N. S., Ispriyanti, D., & Widiharih, T. (2022). Implementasi algoritma fuzzy C-means dan fuzzy possibilistics C-means untuk klasterisasi data tweets pada akun Twitter Tokopedia. Jurnal Gaussian, 11(1), 86–98. https://doi.org/10.14710/j.gauss.v11i1.33996

Ren, F., Zhang, R., & Suganthan, P. N. (2018). Stochastic dominance portfolio optimization using metaheuristic algorithms. Applied Soft Computing, 72, 477–490. https://doi.org/10.3390/a11050072

Sarjana, S., Nardo, R., Hartono, R., Siregar, Z. H., Irmal, Sohilauw, M. I., Wahyuni, S., Rasyid, A., Djaha, Z. A., & Badrianto, Y. (2022). Manajemen risiko. Media Sains Indonesia.

Soleimani, H., Golmakani, H. R., & Salimi, M. H. (2009). Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints, and regarding sector capitalization using genetic algorithm. Expert Systems with Applications, 36(3), 5058–5063. https://doi.org/10.1016/j.eswa.2008.06.007

Suhirman, S., & Wintolo, H. (2019). System for determining public health level using the agglomerative hierarchical clustering method. Compiler, 8(1), 95–104. https://doi.org/10.28989/compiler.v8i1.425

Wahyuni, R., Mahmudy, W. F., & Setiawan, B. D. (2017). Penentuan portofolio saham optimal menggunakan algoritma genetika. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 1(1), 63–68.

Xiong, Y., Zou, Z., & Cheng, J. (2023). Cuckoo search algorithm based on cloud model and its application. Scientific Reports, 13, 10098. https://doi.org/10.1038/s41598-023-37326-3

Yang, X.-S., & He, X. (2013). Firefly algorithm: Recent advances and applications. International Journal of Swarm Intelligence, 1(1), 36–50. https://doi.org/10.1504/IJSI.2013.055801