BAB I
PENDAHULUAN

1.1 Latar Belakang

Investor melakukan investasi dengan harapan memperoleh keuntungan semaksimum mungkin di masa depan dengan risiko sekecil mungkin. Akan tetapi, dalam praktisnya, semakin tinggi potensi keuntungan, semakin tinggi pula risikonya. Oleh karena itu, diperlukan strategi yang tepat untuk mengurangi risiko. Prinsip yang umum diterapkan adalah mengalokasikan modal pada lebih dari satu aset. Kumpulan aset yang terpilih disatukan membentuk sebuah portofolio (Tandelilin, 2010).

Portofolio diartikan sebagai serangkaian kombinasi beberapa aset yang diinvestasi dan dipegang oleh investor, baik perorangan maupun lembaga. Setiap aset yang terdapat di pasar modal memiliki karakteristiknya masing-masing. Oleh karena itu, setiap investor perlu membuat portofolio yang sesuai dengan preferensinya terhadap tingkat *return* dan risiko. Kondisi ini menjadikan perlunya sebuah kerangka kerja yang sistematis dan kuantitatif untuk mengelola hubungan fundamental antara tingkat risiko dan *return* secara efektif (Sunariyah, 2011).

Tantangan tersebut dijawab oleh Harry Markowitz yang memperkenalkan Teori Portofolio Modern pada tahun 1952, sebuah solusi yang menjadi acuan hingga saat ini. Gagasan fundamental Markowitz adalah risiko sebuah aset investasi tidak semestinya dievaluasi secara terisolasi, melainkan perlu dilihat dari kontribusinya terhadap risiko portofolio secara keseluruhan. Kunci utamanya terletak pada diversifikasi, yakni mengombinasikan berbagai aset yang tidak berkorelasi sempurna satu sama lain (Markowitz, 1952).

Pembuatan portofolio terus berkembang seiring berjalannya waktu. Kondisi tingkat *return* aset yang tidak terdistribusi normal secara umum mengakibatkan model portofolio yang diajukan oleh Markowitz pada tahun 1952 tidak tepat untuk digunakan. Hal ini dibuktikan dengan penelitian Athayde dan Flôres serta Jondeau dan Rockinger yang menjelaskan bahwa *return* aset dengan distribusi yang sangat menyimpang akan membuat hasil pembobotan portofolio kurang memuaskan (Athayde dan Flôres, 2004; Jondeau dan Rockinger, 2006). Oleh karena itu, dilakukan sejumlah metode menggunakan *machine learning* yang dikelompokkan berdasarkan fungsinya menjadi *supervised learning, unsupervised learning,* dan *reinforcement learning* (Morales dan Escalante, 2022). Mayoritas data mentah memiliki ukuran yang besar dan tidak memiliki label, sehingga untuk menganalisisnya diperlukan metode *unsupervised learning* (Naeem et al., 2023).

Unsupervised learning merupakan salah satu metode *machine learning* yang menganalisis data tidak berlabel dan berfungsi untuk mengidentifikasi pola atau hubungan tersembunyi dalam data. Salah satu metode *unsupervised learning* adalah *clustering*, yaitu pengelompokan data berdasarkan kesamaan yang dimiliki sehingga diperoleh sejumlah klaster dalam data (Ebrahimi et al., 2025). Rodriguez beserta koleganya menjelaskan bahwa *clustering* diharapkan dapat menempatkan setiap informasi data ke dalam satu klaster saja (Rodriguez et al., 2019). Dalam kasus pengelompokan data saham, telah dilakukan berbagai macam algoritma *clustering*. Misalnya pengelompokan berdasarkan rasio kinerja perusahaan menggunakan HDBSCAN (Jearanaitanakij et al., 2025), ataupun berdasarkan valuasi di pasar modal menggunakan *K-Means* (Pratama et al., 2024). Metode *clustering* ini juga sangat membantu dalam pengelompokan aset-aset investasi selain saham seperti *cryptocurrency* (Bulani et al., 2025).

Penelitian yang dilakukan oleh León dan koleganya berhasil menunjukkan bahwa portofolio yang dibuat berdasarkan analisis *clustering* mampu mengalahkan kinerja portofolio *mean-variance* dalam investasi jangka panjang. Hal ini dikarenakan volatilitas portofolio *mean-variance* yang jauh lebih tinggi daripada portofolio hasil analisis *clustering* seperti *hierarchical clustering* dan *K-Means*, serta tingkat *return* yang semakin menurun seiring berjalannya waktu (León et al., 2017). Hasil *backtesting* dari penelitian Tayali juga menunjukkan bahwa portofolio *mean-variance* hasil analisis *clustering* bekerja lebih baik ketimbang portofolio *mean-variance* yang tidak menggunakan pendekatan *clustering* (Tayali, 2020).

Metode *clustering* sejatinya hanya membantu dalam pengelompokan aset, tetapi tidak dalam pembobotannya. Oleh karena itu, diperlukan pula algoritma optimasi yang dapat melakukannya. Salah satu kelompok algoritma optimasi adalah algoritma meta-heuristik yang terinspirasi dari cara alam bekerja (*nature-based*) (Tomar et al., 2023). Beberapa algoritma metaheuristik yang terinspirasi dari alam adalah algoritma *grey wolf*, algoritma *bat*, algoritma *whale*, algoritma *ant lion colony*, algortima genetik, dan algoritma *particle swarm* (Tomar et al., 2023; Setiawan dan Rosadi, 2019). Kelebihan dari algoritma ini adalah kebolehannya untuk penggabungan satu algoritma dan algoritma lainnya, menciptakan sebuah algoritma gabungan (*hybrid*) dengan kinerja optimasi yang lebih bagus (Tomar et al., 2023).

Penelitian Farid dan Rosadi memadukan algoritma *clustering* berupa *self-organizing maps* (SOM) untuk pemilihan saham, dan algoritma genetik untuk pembobotan sahamnya. Hasil penelitiannya menunjukkan bahwa portofolio yang dibuat memiliki kinerja yang sedikit lebih baik daripada portofolio Markowitz hasil analisis *clustering* (Farid dan Rosadi, 2022). Beberapa penelitian sebelumnya menunjukkan bahwa algoritma optimasi metaheuristik dapat dimodifikasi sehingga menghasilkan portofolio aset yang lebih optimal daripada portofolio buatan metode metaheuristik biasa (Anam et al., 2025; Bulani et al., 2025). Selain itu, penelitian Kuo dan Hong menunjukkan bahwa algoritma metaheuristik gabungan

menghasilkan portofolio reksadana yang lebih baik daripada metode metaheuritstik biasa (Kuo dan Hong, 2013).

Penelitian ini mengadopsi pendekatan dua lapis: pertama mengelompokkan saham-saham menggunakan algoritma *clustering* yang terbaik, dan kedua, melakukan pembobotan saham-saham perwakilan tiap klaster menggunakan gabungan algoritma genetika dan *particle swarm optimization*, lalu mengukur kinerja portofolio yang dihasilkan.

1.2 Rumusan Masalah

Permasalahan utama dalam penelitian ini dapat dirumuskan dalam pertanyaan penelitian sebagai berikut:

1. Bagaimana pengelompokan saham berdasarkan algoritma klaster terbaik dan menentukan saham perwakilan setiap klaster?
2. Bagaimana kinerja portofolio saham perwakilan klaster yang dibuat menggunakan gabungan algoritma genetika dan *particle swarm optimization*?

1.3 Batasan Masalah

Ruang lingkup penelitian dibatasi sebagai berikut:

1. Objek penelitian: Penelitian ini terbatas pada saham-saham Indeks LQ-45. Mengingat sifat dinamis dari keanggotaan indeks, proses seleksi dilakukan untuk mengidentifikasi hanya saham-saham yang secara konsisten terdaftar selama keseluruhan periode 1 Januari 2022 sampai 31 Desember 2024.
2. Data: Data yang digunakan adalah data harga penutupan harian (*daily closing price*) selama periode penelitian dan laporan keuangan tahunan terbaru yang bersumber dari situs data keuangan publik (*Yahoo Finance*).
3. Model Optimasi: Pembentukan portofolio optimal hanya menggunakan algoritma klaster terbaik dari *K-Means*, *Agglomerative Hierarchy*, dan DBSCAN dengan kriteria terbaik berupa *Silhouette Score*. Selanjutnya dilakukan pembobotan saham menggunakan gabungan algoritma genetika dan *Particel Swarm Optimization*. Proses ini dijalankan secara komputasional menggunakan *RStudio* dengan kriteria optimasi berupa Rasio Sharpe.
4. Asumsi *Risk-Free Rate* (RFR): Penelitian ini menggunakan nilai untuk tingkat imbal hasil bebas risiko sebesar 6.00%. Angka ini diambil dari suku bunga acuan Bank Indonesia (BI7DRR) pada akhir periode penelitian.

1.4 Tujuan Penelitian

Sejalan dengan rumusan masalah yang telah ditetapkan, tujuan dari penelitian ini adalah:

1. Mengelompokkan saham-saham menggunakan algoritma klaster terbaik dan menentukan saham perwakilan setiap klaster.
2. Menghasilkan portofolio saham optimal berdasarkan hasil analisis klaster menggunakan gabungan algoritma genetika dan *particle swarm optimization* dan mengukur kinerjanya.

1.5 Manfaat Penelitian

Hasil dari penelitian ini diharapkan bermanfaat bagi berbagai pihak, antara lain:

1. Bagi Investor dan Manajer Investasi: Memberikan sebuah model portofolio optimal yang konkret dan berbasis data empiris sebagai referensi strategi investasi di pasar saham Indonesia.
2. Bagi Akademisi dan Peneliti Selanjutnya: Menjadi studi empiris penerapan algoritma klaster dan metaheuristik pada data pasar saham Indonesia terkini (era pasca-pandemi).
3. Bagi Penulis: Sebagai pemenuhan salah satu syarat untuk memperoleh gelar sarjana pada program studi terkait.

1.6 Tinjauan Pustaka

Investasi adalah suatu komitmen atas sejumlah dana atau aset-aset lainnya, baik yang berwujud maupun yang tidak berwujud, pada masa sekarang untuk memperoleh keuntungan di masa yang akan datang (Dewi, 2021). Ketika berinvestasi, keahlian dalam memprediksi imbal hasil (*return*) menjadi sangat penting. Prediksi mengenai *return* di masa depan tidak akan lepas dari yang namanya ketidakpastian. Mengingat imbal hasil bisa jadi bernilai positif (untung) atau bernilai negatif (rugi) (Fittria dan Zaenal, 2024).

Investor selalu mengharapkan *return* yang optimal, tetapi sering kali terjadi perbedaan antara *return* yang diharapkan (*expected return*) dengan *return* sebenarnya yang diterima. Hal ini memperlihatkan bahwa investor tidak mampu memprediksi secara akurat besaran imbal hasil yang akan diperoleh, sehingga setiap investasi selalu disertai dengan adanya risiko (*risk*) (Fittria dan Zaenal, 2024). Tingkat risiko dan ekspektasi *return* memiliki korelasi yang positif, artinya semakin besar risiko suatu aset investasi, semakin besar pula *return* yang diharapkan, (*high yield means high risk*), dan juga sebaliknya (Dewi, 2021). Apabila diurutkan berdasarkan tingkat risikonya, aset seperti opsi memiliki tingkat risiko lebih tinggi ketimbang waran, sedangkan waran memiliki tingkat risiko lebih besar

daripada aset yang umum diinvestasikan seperti saham maupun obligasi (Jogiyanto, 2017).

Keputusan dalam berinvestasi umumnya dipengaruhi oleh tiga hal. Pertama, imbal hasil yang diharapkan dari aset yang diinvestasikan. Kedua, hasil evaluasi risiko yang dihadapi, dikarenakan adanya hubungan yang selaras antara potensi keuntungan dan tingkat risiko. Terakhir, jangka waktu investasi yang menjadi tujuan finansial, apakah itu jangka pendek, menengah, atau panjang. Seorang investor yang ulung tidak hanya berfokus pada *return* yang tinggi, melainkan manajemen risiko yang efektif. Oleh karena itu, untuk menjaga keseimbangan antara potensi keuntungan dan pengelolaan risiko yang menjadi kunci keberhasilan investasi jangka panjang, membentuk portofolio aset dapat menjadi solusinya (Fittria dan Zaenal, 2024).

Portofolio merupakan sekumpulan aset-aset investasi yang mencakup saham, obligasi, reksadana, deposito, komoditas, tanah, karya seni, atau benda lainnya yang memiliki nilai (Koch et al., 2008). Dengan berinvestasi pada berbagai aset, seorang investor dapat berfokus pada sejumlah aset ketika beberapa aset lainnya mengalami penurunan imbal hasil. Dalam menyusun portofolio, investor perlu mengatur kombinasi aset beserta bobotnya masing-masing sehingga dapat meminimalkan risiko dan memaksimalkan *expected return*, sehingga terbentuk suatu portofolio yang optimal (Fittria dan Zaenal, 2024).

1.6.1 Rasio Keuangan

Investor memiliki berbagai cara untuk menilai laporan keuangan suatu perusahaan, salah satunya dengan menggunakan rasio keuangan (*finantial ratio*). Teknik ini lazim digunakan oleh para investor maupun manajer keuangan dikarenakan menggunakan rumus perhitungan yang sederhana, serta akses laporan keuangan yang terbuka untuk umum. Oleh karena itu, rasio keuangan tidak hanya digunakan untuk membandingkan laporan keuangan perusahaan dari tahun ke tahun, melainkan juga untuk membandingkan kinerja dua atau lebih perusahaan berbeda dalam satu periode (Ross et al., 2022).

Rasio-rasio keuangan dikelompokkan berdasarkan aspek performa perusahaan yang ingin diinvestigasi. Rasio profitabilitas merupakan ukuran kinerja perusahaan dalam menghasilkan laba dari aset yang dimiliki. Rasio aktivitas mengukur kinerja perusahaan dalam mengelola berbagai aset yang dimiliki, seperti piutang dan persediaan barang. Rasio *leverage* adalah ukuran kinerja perusahaan dalam menghadapi kewajiban jangka panjang, misalnya utang jangka panjang. Rasio likuiditas berfungsi untuk mengukur kemampuan perusahaan dalam memenuhi kewajiban jangka pendeknya, seperti pembayaran pajak dan gaji karyawan. Rasio valuasi mengukur kualitas aset atau aliran dana perusahaan yang berkaitan erat dengan surat kepemilikan perusahaan, contohnya saham (Robinson et al., 2015). Penelitian ini akan berfokus pada rasio *Return-On-Equity* sebagai

salah satu rasio profitabilitas, *Debt-to-Equity* sebagai salah satu rasio *leverage*, dan *Cash ratio* sebagai salah satu rasio likuiditas.

a. *Return On Equity*

Salah satu hal dari perusahaan yang paling sering diperhatikan, khususnya oleh investor, adalah kemampuannya dalam menghasilkan keuntungan dari modal yang diberikan. Profitabilitas suatu perusahaan sangat memengaruhi valuasi saham-sahamnya yang diterbitkan di pasar modal. Tidak hanya itu, kemampuan dalam menghasilkan profit juga dapat menggambarkan kualitas manajemen di perusahaan tersebut (Robinson et al., 2015).

Salah satu ukuran profitabilitas perusahaan yang paling umum digunakan adalah perbandingan pendapatan bersihnya terhadap modal yang dimiliki atau *Return-On-Equity*. Hal ini dikarenakan apabila pendapatan bersihnya lebih sedikit daripada modal yang ditanam, maka investor akan ragu untuk terus menanamkan modalnya karena dianggap kurang menguntungkan. Perhitungan *Return-On-Equity* dapat diformulasikan sebagai berikut (Ross et al., 2022):

$$ROE = \frac{Net\ Income}{Shareholder's\ Equity} \tag{1}$$

b. *Debt-to-Equity*

Perusahaan-perusahaan pada umumnya tidak akan terlepas dari yang namanya utang. Akan tetapi, utang yang dimiliki mesti diatur sedemikian rupa sehingga tidak membebani perusahaan di masa depan. Ukuran yang digunakan untuk menggambarkan kemampuan perusahaan dalam melunasi utangnya di masa depan dikenal sebagai *leverage ratio* (Robinson et al., 2015). Salah satu *leverage ratio* yang sering digunakan adalah rasio *debt-to-equity*, yakni perbandingan utang perusahaan terhadap jumlah modal yang dimiliki. Rumus dari *Debt-to-Equity ratio* dapat dinyatakan sebagai berikut (Berk dan Peter, 2024):

$$Debt\ to\ Equity = \frac{Total\ Debt}{Shareholder's\ Equity} \tag{2}$$

Berdasarkan rumus di Persamaan (2), dapat diketahui bahwa semakin besar rasio *Debt-to-Equity*, maka semakin besar pula risiko yang dimiliki sebuah perusahaan. Hal ini dikarenakan sebagian besar aset perusahaan tersebut didominasi oleh utang, khususnya utang jangka panjang. Kondisi tersebut menimbulkan kekhawatiran akan gagalnya perusahaan dalam melunasi utang-utangnya di masa depan.

c. *Cash Ratio*

Cash rasio merupakan salah satu rasio likuiditas. Rasio likuiditas sendiri merupakan ukuran yang menggambarkan kemampuan sebuah perusahaan untuk membayar kewajiban jangka pendeknya. Hal ini dapat terlihat dari banyaknya

aktiva lancar yang bisa dikonversikan menjadi uang tunai dalam waktu dekat. Dengan demikian, rasio likuiditas berfokus pada arus kas perusahaan (Robinson et al., 2015). Nilai dari *cash ratio* bisa diperoleh menggunakan rumus berikut (Berk dan Peter, 2024):

$$Cash\ Ratio = \frac{Cash}{Current\ Liabilities} \tag{3}$$

Cash ratio digunakan untuk menggambarkan kemampuan dalam melunasi kewajiban jangka pendek, khususnya di saat krisis (Robinson et al., 2015). Semakin besar *cash ratio* yang dimiliki perusahaan, maka semakin mudah perusahaan tersebut dalam mencairkan dananya. Rasio ini juga kerap digunakan oleh kreditur yang ingin meminjamkan dananya kepada perusahaan dalam jangka pendek (Ross et al., 2022).

1.6.2 Nilai *Eigen* dan Vektor *Eigen*

Definisi 1.15: *(Anton dan Kaul, 2019) Jika* $\mathbf{\Sigma}$ *merupakan matriks berukuran* $p \times p$*, maka sebuah vektor tak nol* $\mathbf{x}$ *di* $\mathbb{R}^p$ *dinamakan vektor eigen dari* $\mathbf{\Sigma}$ *jika memenuhi persamaan*

$$\mathbf{\Sigma x} = \lambda \mathbf{x} \tag{4}$$

untuk suatu skalar λ*. Skalar* λ *ini dinamakan nilai eigen dari* $\mathbf{\Sigma}$ *dan* $\mathbf{x}$ *disebut sebagai vektor eigen yang berkorespondensi dengan* λ*.*

Berdasarkan definisi di atas, operasi perkalian matriks $\mathbf{\Sigma}$ dengan vektor $\mathbf{x}$ hanya mengubah panjang dari vektor $\mathbf{x}$ atau membalikkan arah vektornya.

Teorema 1.7: *(Anton dan Kaul, 2019) Diberikan matriks* $\mathbf{\Sigma}$ *berukuran* $p \times p$*, pernyataan berikut saling ekuivalen satu sama lain*

a. λ *adalah nilai eigen dari* $\mathbf{\Sigma}$
b. λ *adalah solusi dari persamaan* $det(\lambda \boldsymbol{I} - \mathbf{\Sigma}) = 0$
c. *Sistem persamaan* $(\lambda \boldsymbol{I} - \mathbf{\Sigma})\mathbf{x} = \mathbf{0}$ *memiliki solusi non-trivial*
d. *Terdapat vektor tak nol* $\mathbf{x}$ *yang memenuhi* $\mathbf{\Sigma x} = \lambda \mathbf{x}$

1.6.3 Analisis Multivariat

Analisis multivariat merupakan metode analisis dua atau lebih variabel dari suatu data secara simultan. Tujuan utamanya adalah untuk mengetahui struktur setiap variabel dan hubungan antar variabel.

1.6.3.1 Matriks Data Multivariat

Diberikan n objek penelitian dan p variabel data, sehingga diperoleh sebanyak np informasi. Dengan demikian, sampel data analisis multivariat dapat dinyatakan sebagai berikut:

		Variabel ke-1	Variabel ke-2	⋯	Variabel ke-p
		x_1	x_2	⋯	x_p
Objek ke-1	o_1	x_{11}	x_{12}	⋯	x_{1p}
Objek ke-2	o_2	x_{21}	x_{22}	⋯	x_{2p}
⋮	⋮	⋮	⋮	⋱	⋮
Objek ke-n	o_n	x_{n1}	x_{n2}	⋯	x_{np}

Berdasarkan bentuk di atas, sampel data multivariat dapat dinyatakan dalam bentuk matriks $\boldsymbol{X}$ berukuran $n \times p$. Baris dipandang sebagai objek penelitian dan kolom dapat dipandang sebagai variabel, sehingga menjadi matriks sebagai berikut:

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & & x_{2p} \\ \vdots & & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}_{n \times p} \tag{5}$$

dengan x_{ij} menyatakan data dari objek ke-i pada variabel ke-j, n dan p berturut-turut adalah jumlah baris dan jumlah kolom dari $\boldsymbol{X}$.

1.6.3.2 Vektor Rata-Rata dan Matriks Variansi-Kovariansi

Vektor acak atau matriks acak merupakan vektor atau matriks yang memiliki entri berupa variabel acak. Oleh karena itu, nilai harapan dari sebuah vektor acak $\boldsymbol{y}$ berukuran $p \times 1$ didefinisikan sebagai berikut (Rencher dan Schaalje, 2008):

$$E[\boldsymbol{y}] = E\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix} = \begin{bmatrix} E[y_1] \\ E[y_2] \\ \vdots \\ E[y_p] \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix} = \boldsymbol{\mu} \tag{6}$$

dengan $y_1, y_2, \ldots, y_p$ adalah variabel acak dari vektor $\boldsymbol{y}$ dan $\mu_i = E[y_i]$ untuk $i = 1,2, \ldots, p$. Lebih lanjut, jika diberikan vektor acak vektor $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$ dan $\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix}$, maka berdasarkan operasi penjumlahan vektor dan sifat dari ekspektasi, diperoleh

$$E[\boldsymbol{x} + \boldsymbol{y}] = E[\boldsymbol{x}] + E[\boldsymbol{y}] \tag{7}$$

Sementara itu, misalkan σ_{ij} sebagai kovariansi dari variabel acak y_i dan y_j, dengan $i, j = 1,2, \ldots, p$. Matriks variansi-kovariansi dari vektor acak $\boldsymbol{y}$, yang dinotasikan dengan $\boldsymbol{\Sigma}$, didefinisikan sebagai berikut (Rencher dan Schaalje, 2008):

$$\boldsymbol{\Sigma} = Cov(\boldsymbol{y}) = \left[\sigma_{ij}\right]_{p\times p} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & & \sigma_{2p} \\ \vdots & & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp} \end{bmatrix}_{p\times p} \tag{8}$$

Proses pembentukan matriks variansi-kovariansi dari vektor acak $\boldsymbol{y}$ sebagai berikut:

$\boldsymbol{\Sigma} = Cov(\boldsymbol{y})$

$\boldsymbol{\Sigma} = E[(\boldsymbol{y} - \boldsymbol{\mu})(\boldsymbol{y} - \boldsymbol{\mu})^T]$

$\boldsymbol{\Sigma} = E[\boldsymbol{y}\boldsymbol{y}^T - \boldsymbol{y}\boldsymbol{\mu}^T - \boldsymbol{\mu}\boldsymbol{y}^T + \boldsymbol{\mu}\boldsymbol{\mu}^T]$

$\boldsymbol{\Sigma} = E[\boldsymbol{y}\boldsymbol{y}^T] - E[\boldsymbol{y}\boldsymbol{\mu}^T] - E[\boldsymbol{\mu}\boldsymbol{y}^T] + E[\boldsymbol{\mu}\boldsymbol{\mu}^T]$

$$\boldsymbol{\Sigma} = \begin{bmatrix} E[(y_1-\mu_1)(y_1-\mu_1)] & E[(y_1-\mu_1)(y_2-\mu_2)] & \cdots & E\left[(y_1-\mu_1)(y_p-\mu_p)\right] \\ E[(y_2-\mu_2)(y_1-\mu_1)] & E[(y_2-\mu_2)(y_2-\mu_2)] & & E\left[(y_2-\mu_2)(y_p-\mu_p)\right] \\ \vdots & & \ddots & \vdots \\ E\left[(y_p-\mu_p)(y_1-\mu_1)\right] & E\left[(y_p-\mu_p)(y_2-\mu_2)\right] & \cdots & E\left[(y_p-\mu_p)(y_p-\mu_p)\right] \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & & \sigma_{2p} \\ \vdots & & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp} \end{bmatrix}$$

Berdasarkan definisinya, ditambah dengan salah satu sifat kovariansi, yakni $Cov(y_i, y_j) = Cov(y_j, y_i)$ untuk $i, j = 1,2, \ldots p$, matriks variansi-kovariansi tergolong simetris. Selain itu, matriks variansi-kovariansi tergolong ke dalam matriks semi-definit positif, yakni matriks dengan semua nilai *eigen*-nya adalah bilangan riil non-negatif.

Selanjutnya untuk kasus sampel, vektor mean atau rata-rata dinotasikan dengan $\bar{\boldsymbol{x}}$, sedangkan matriks variansi-kovariansinya dinotasikan dengan $\boldsymbol{S}$. Vektor mean sampel yang terdiri atas p variabel dapat ditulis sebagai berikut:

$$\bar{\boldsymbol{x}} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_p \end{bmatrix} \tag{9}$$

Sementara itu, kovariansi antara variabel ke-i dengan variabel ke-j untuk sampel dihitung menggunakan rumus berikut:

$$S_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j) \tag{10}$$

dengan $i, j = 1,2, \ldots, p$. Dengan demikian, matriks variansi-kovariansi dapat ditulis sebagai berikut:

$$\boldsymbol{S} = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1p} \\ S_{21} & S_{22} & & S_{2p} \\ \vdots & & \ddots & \vdots \\ S_{p1} & S_{p2} & \cdots & S_{pp} \end{bmatrix} \tag{11}$$

Berdasarkan penulisan pada Persamaan (11), jelas bahwa matriks $\boldsymbol{S}$ tergolong simetris.

1.6.3.3 Kombinasi Linear Variabel Acak

Kombinasi linear dari kumpulan variabel acak $x_1, x_2, \ldots x_n$ adalah sebuah variabel acak $Y = w_1x_1 + w_2x_2 + \cdots + w_nx_n$, dengan w_i merupakan suatu konstan untuk $i = 1,2,\ldots,n$. Apabila diberikan vektor acak $\boldsymbol{x}^T = [x_1 \quad x_2 \quad \cdots \quad x_n]$ dan vektor konstan $\boldsymbol{w}^T = [w_1 \quad w_2 \quad \cdots \quad w_n]$, maka hasil kombinasi linearnya dapat dinyatakan sebagai berikut:

$$Y = \boldsymbol{w}^T\boldsymbol{x} = [w_1 \quad w_2 \quad \cdots \quad w_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{12}$$

Berdasarkan Persamaan (12), nilai harapan dari variabel acak Y adalah sebagai berikut:

$$E[Y] = E[w_1x_1 + w_2x_2 + \cdots + w_nx_n]$$

$$E[Y] = E[w_1x_1] + E[w_2x_2] + \cdots + E[w_nx_n]$$

$$E[Y] = w_1E[x_1] + w_2E[x_2] + \cdots + w_nE[x_n]$$

$$E[Y] = [w_1 \quad w_2 \quad \cdots \quad w_n] \begin{bmatrix} E[x_1] \\ E[x_2] \\ \vdots \\ E[x_n] \end{bmatrix}$$

$$E[Y] = [w_1 \quad w_2 \quad \cdots \quad w_n] \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}$$

$$E[Y] = \boldsymbol{w}^T\boldsymbol{\mu} \tag{13}$$

Lebih lanjut, variansi dari variabel acak Y dapat dituliskan sebagai berikut:

$$Var(Y) = Var(w_1x_1 + w_2x_2 + \cdots + w_nx_n)$$

$$Var(Y) = \sum_{i=1}^{n} Var(w_ix_i) + 2\sum_{1\le j<k\le n} w_jw_kCov(x_j, x_k)$$

$$Var(Y) = \sum_{i=1}^{n} w_i^2\sigma_{ii} + 2\sum_{1\le j<k\le n} w_jw_k\sigma_{jk}$$

$$Var(Y) = w_1(w_1\sigma_{11} + w_2\sigma_{12} + \cdots + w_n\sigma_{1n}) + \cdots + w_n(w_1\sigma_{n1} + w_2\sigma_{n2} + \cdots + w_n\sigma_{nn})$$

$$Var(Y) = [w_1 \quad w_2 \quad \cdots \quad w_n] \begin{bmatrix} (w_1\sigma_{11} + w_2\sigma_{12} + \cdots + w_n\sigma_{1n}) \\ (w_1\sigma_{21} + w_2\sigma_{22} + \cdots + w_n\sigma_{2n}) \\ \vdots \\ (w_1\sigma_{n1} + w_2\sigma_{n2} + \cdots + w_n\sigma_{nn}) \end{bmatrix}$$

$$Var(Y) = [w_1 \quad w_2 \quad \cdots \quad w_n] \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & & \sigma_{2n} \\ \vdots & & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{nn} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

$$Var(Y) = \boldsymbol{w}^T \boldsymbol{\Sigma} \mathbf{w} \tag{14}$$

dengan $\boldsymbol{\Sigma}$ merupakan matriks variansi-kovariansi dari vektor acak $\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$.

1.6.4 Saham

Saham merupakan tanda penyertaan atau kepemilikan seseorang atau organisasi dalam suatu perusahaan (Fadilah et al., 2023). Saham berupa selembar kertas yang menerangkan bahwa pemilik kertas tersebut adalah pemilik dari suatu perusahaan yang menerbitkan lembaran saham tersebut. Sederhananya, saham dapat diidentifikasikan sebagai sertifikat kepemilikan suatu perusahaan dan pemegangnya memiliki hak klaim atas penghasilan dan aset perusahaan. Imbalan yang akan diperoleh dari kepemilikan saham berupa dividen atau *capital gain* (Fadila dan Nuswandari, 2022). Dividen merupakan pembagian keuntungan yang diberikan oleh perusahaan dan bersumber dari keuntungan yang dihasilkan oleh perusahaan. Dividen dapat berupa uang tunai atau sejumlah saham dan diberikan setelah mendapat persetujuan dari pemegang saham dalam Rapat Umum Pemegang Saham (RUPS). Sedangkan *capital gain* merupakan hasil selisih antara harga beli dengan harga jual (Mar'ati, 2010).

Harga saham merupakan harga satu lembar saham yang dijual di pasar modal. Harga saham merupakan salah satu indikator penting yang diperhatikan para investor ketika mau berinvestasi. Hal ini dikarenakan harga saham menunjukkan kualitas manajemen bisnis dari sebuah perusahaan (Fadila dan Nuswandari, 2022). Harga saham yang tinggi mengindikasikan kondisi perusahaan yang baik, sehingga banyak investor yang berbondong-bondong memperdagangkan saham perusahaan tersebut di pasar modal (Fatikhah dan Puryandani, 2020).

Harga saham dalam satu periode perdagangan sekuritas dibagi menjadi empat macam, yakni harga tertinggi (*high price*), harga terendah (*low price*), harga pembukaan (*open price*), dan harga penutupan (*closing price*). Harga tertinggi atau terendah adalah harga tertinggi tinggi atau terendah yang terjadi dalam satu periode perdagangan. Sementara itu, harga pembuka atau penutup adalah harga

yang terjadi pada awal atau akhir periode perdagangan sekuritas (Fadilah et al., 2023).

Praktisnya, harga saham di pasar modal selalu berubah, baik meningkat atau menurun. Hal ini sangat dipengaruhi oleh banyaknya permintaan dan penawaran saham. Akibatnya, tingkat imbal hasil (*return*) yang diperoleh pada setiap periode perdagangan akan selalu berubah-ubah seiring berjalannya waktu. Perhitungan tingkat *return* dalam skripsi ini berfokus pada penggunaan *realized return* yang memiliki rumus sebagai berikut (Widyaningrum et al., 2022):

$$(R_i)_t = \frac{(P_i)_t - (P_i)_{t-1}}{(P_i)_{t-1}} \tag{16}$$

dengan $(R_i)_t$ merupakan tingkat *return* saham ke-i pada periode ke-t, $(P_i)_t$ merupakan harga saham ke-i pada periode ke-t, dan $(P_i)_{t-1}$ merupakan harga saham ke-i pada periode ke-$(t-1)$. Lebih lanjut, berdasarkan rumus di atas, dapat diketahui bahwa: (a) $(R_i)_t < 0$ jika $(P_i)_t < (P_i)_{t-1}$; (b) $(R_i)_t = 0$ jika $(P_i)_t = (P_i)_{t-1}$; dan (c) $(R_i)_t > 0$ jika$(P_i)_t > (P_i)_{t-1}$.

1.6.4.1 *Expected Return* dan Risiko Saham

Expected return merupakan besar pengembalian yang diharapkan investor di masa depan. *Expected return* dapat diartikan sebagai proyeksi tingkat imbal hasil pada masa mendatang. Nilai *expected return* dapat menjadi acuan bagi investor dalam memprediksi tingkat imbal hasil suatu saham, tetapi prediksi tersebut tidak sepenuhnya akurat dikarenakan potensi munculnya kejadian-kejadian tidak terduga atau perubahan sentimen pasar (Jogiyanto, 2017). Nilai dari *expected return* saham selama T periode dapat dirumuskan sebagai berikut:

$$E[R]_i = \frac{1}{T}\sum_{t=1}^{T}(R_i)_t \tag{17}$$

dengan $(R_i)_t$ sebagai tingkat *return* saham pada periode ke-t dan T merupakan jumlah periode yang diobservasi.

Risiko saham diartikan sebagai besar pergeseran valuasi saham di pasar modal dari nilai yang diharapkan pada jangka waktu tertentu. Risiko saham meliputi kemungkinan kerugian finansial yang berpotensi dialami oleh investor sebagai akibat dari perubahan harga saham yang tidak sesuai harapan. Beberapa faktor yang menyebabkan risiko saham antara lain perubahan kondisi geopolitik, perubahan kebijakan internal, ketidakpastian pasar, perubahan tren, dan faktor-faktor lain yang memengaruhi perusahaan atau pasar secara menyeluruh (Santoso et al., 2023).

Risiko merupakan probabilitas terjadinya penyimpangan atas harapan tingkat pengembalian investasi di masa depan dan diukur menggunakan metode-metode

statistika (Tandelilin, 2017). Salah satu ukuran yang bisa digunakan untuk mengukur risiko saham adalah standar deviasi *return* saham. Standar deviasi *return* saham merupakan akar kuadrat dari variansi *return* saham yang diformulasikan sebagai berikut:

$$\sigma_i = \sqrt{Var(R)_i} = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}((R_i)_t - E[R]_i)^2} \tag{18}$$

1.6.5 Teori Portofolio Modern

Konsep diversifikasi portofolio merupakan ide fundamental dalam dunia keuangan yang diperkenalkan pertama kali oleh Harry Markowitz pada tahun 1950-an. Ide yang disebut sebagai *modern portfolio theory* (MPT) ini berfokus pada meminimalkan risiko investasi dengan mendistribusikan modal pada beberapa kelompok aset. Tujuan utama dari MPT adalah mengurangi tingkat risiko pada tingkat imbal hasil tertentu, atau memaksimalkan tingkat *return* pada tingkat risiko tertentu. Dengan demikian, portofolio yang dibentuk mampu memberikan keseimbangan terbaik antara risiko dan imbal hasil (Fabozzi et al., 2002).

Asumsi utama dari pengembalian MPT adalah tingkat *return* setiap aset yang diinvestasikan memiliki distribusi normal. Dengan demikian, pengukuran kinerja model portofolio Markowitz hanya bergantung pada mean dan *variance*. Mean merupakan ukuran tingkat *return* yang diharapkan, sedangkan *variance* (variansi) adalah ukuran tingkat risiko. Akan tetapi, dalam praktisnya, tingkat *return* aset secara umum tidak terdistribusi secara normal, khususnya saat terjadi krisis keuangan yang sering dipicu oleh kondisi geopolitik ataupun keputusan sejumlah perusahaan ternama (Omisore et al., 2012).

Portofolio yang terdiri atas N aset memiliki besaran bobot pada setiap asetnya yang dinotasikan w_i sebagai bobot aset ke-i, dengan $i = 1,2,3,\dots,N$. Setiap bobot ini berada dalam interval $[0,1]$ dan memenuhi persamaan

$$\sum_{i=1}^{N} w_i = 1 \tag{19}$$

Bobot-bobot ini akan memengaruhi tingkat imbal hasil portofolio yang akan diperoleh investor, serta tingkat risiko yang akan diterima. Pembobotan aset portofolio yang belum tentu sama rata mengakibatkan tingkat *return*-nya pada periode ke-t menjadi:

$$(R_P)_t = w_1(R_1)_t + w_2(R_2)_t + \cdots + w_{N-1}(R_{N-1})_t + w_N(R_N)_t \tag{20}$$

dengan $(R_i)_t$ merupakan tingkat *return* aset ke-i pada periode ke-t.

Perhitungan nilai harapan *return* portofolio yang terdiri dari N aset dilakukan dengan memerhatikan bobot dari masing-masing asetnya. Nilai dari *expected return* dari portofolio dengan N aset dapat diformulasikan sebagai berikut:

$$E[R]_P = \sum_{i=1}^{N} w_i E[R]_i \tag{21}$$

dengan $E[R]_i$ merupakan nilai harapan dari *return* aset ke-i, untuk $i = 1,2,3,\dots,N$.

Perhitungan risiko portofolio dapat dilakukan dengan memperhitungkan korelasi antar aset melalui kovariansi. Nilai kovariansi didasarkan pada *return* aset dan berguna untuk melihat hubungan pergerakan *return* antara dua aset berbeda. Nilai kovariansi yang positif mengindikasikan pergerakan *return* yang searah, menandakan pergerakan harga dua aset yang ditinjau juga searah. Sedangkan nilai kovariansi negatif memberi sinyal bahwa pergerakan *return* antara dua aset saling berlawanan, sehingga pergerakan harganya juga saling berlawanan. Akan tetapi, jika nilai kovariansinya tepat atau mendekati 0, maka dua aset yang ditinjau tidak memiliki hubungan dalam pergerakan *return* (Bacon, 2021). Kovariansi antara dua aset berbeda dapat dihitung menggunakan rumus berikut:

$$\sigma_{ij} = \frac{1}{(T-1)} \sum_{t=1}^{T} ((R_i)_t - E[R]_i)\left(\left(R_j\right)_t - E[R]_j\right) \tag{22}$$

dengan T menyatakan banyak periode yang diobservasi. Selanjutnya, risiko pada portofolio yang terdiri dari N aset dapat dihitung menggunakan rumus standar deviasi berikut:

$$\begin{aligned} \sigma_P &= \sqrt{Var\left(\sum_{i=1}^{N} w_i R_i\right)} \\ &= \sqrt{Cov\left(\sum_{i=1}^{N} w_i R_i, \sum_{i=1}^{N} w_i R_i\right)} \\ &= \sqrt{\sum_{1 \le i,j \le N} w_i w_j \sigma_{ij}} = \sqrt{\boldsymbol{w}^T \boldsymbol{\Sigma}_{\mathrm{P}} \boldsymbol{w}} \end{aligned} \tag{23}$$

Dengan $\boldsymbol{w}$ sebagai vektor dari bobot setiap aset ($\boldsymbol{w}^T = [w_1 \quad w_2 \quad \cdots \quad w_N]$) dan $\boldsymbol{\Sigma}_{\mathrm{P}}$ sebagai matriks variansi-kovariansi $\left(\boldsymbol{\Sigma}_{\mathrm{P}} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2N} \\ \vdots & & \ddots & \vdots \\ \sigma_{N1} & \sigma_{N2} & \cdots & \sigma_{NN} \end{bmatrix}\right)$ untuk setiap aset di portofolio.

1.6.6 Rasio Sharpe

Pengukuran risiko dan imbal hasil yang sesuai dari sebuah portofolio memungkinkan dilakukan suatu perbandingan tingkat *return* di masa lalu terhadap suatu tingkat risiko (Sharpe, 1966). Salah satu ukuran utama yang paling umum digunakan adalah rasio Sharpe. Rasio Sharpe merupakan hasil perbandingan antara ekspektasi dari *excess return*, yakni selisih imbal hasil antara aset berisiko dengan aset bebas risiko (*risk-free asset*), dengan standar deviasi. Meskipun begitu, nilai dari kedua komponen tersebut tidak diketahui sehingga perlu mengestimasinya secara statistik (Benhamou, 2021).

Expected excess return dan standar deviasi yang digunakan pada rasio Sharpe sebenarnya berdasar pada paradigma model portofolio Markowtiz. Model tersebut mengasumsikan bahwa rata-rata dan simpangan baku dari data historis imbal hasil merupakan statistik cukup untuk mengevaluasi kinerja portofolio investasi. Rasio Sharpe dapat digunakan untuk mengurutkan kinerja portofolio-portofolio investasi yang dibentuk. Nilai rasio yang semakin tinggi menunjukkan kinerja portofolio yang semakin bagus, demikian sebaliknya. Hal ini dikarenakan nilai *excess return* yang akan diperoleh di masa depan jauh lebih besar daripada risiko yang mesti ditanggung (Tandelilin, 2010).

$$Rasio\ Sharpe\ Saham\ ke-i = S_i = \frac{E[R_a]_i - r_f}{(\sigma_a)_i} \tag{24}$$

$$Rasio\ Sharpe\ Portofolio = S_P = \frac{E[R_a]_P - r_f}{(\sigma_a)_P} \tag{25}$$

dengan $E[R_a]_i$ dan $E[R_a]_P$ berturut-turut adalah ekspektasi *return* tahunan saham ke-i dan portofolio, r_f sebagai tingkat *return* aset bebas risiko, dan $(\sigma_a)_i$ dan $(\sigma_a)_P$ berturut-turut sebagai standar deviasi *return* tahunan saham ke-i dan portofolio.

Apabila data *return* yang diberikan dalam bentuk data historis harian, maka nilai harapan tahunannya dapat dihitung dengan pendekatan *compound annual growth rate*. Rumusnya sebagai berikut (Bacon, 2021):

$$E[R_d]_i = (1 + E[R_a]_i)^{\frac{1}{252}} - 1 \Leftrightarrow E[R_a]_i = (1 + E[R_d]_i)^{252} - 1 \tag{26}$$

$$E[R_d]_P = (1 + E[R_a]_P)^{\frac{1}{252}} - 1 \Leftrightarrow E[R_a]_P = (1 + E[R_d]_P)^{252} - 1 \tag{27}$$

dengan $E[R_d]_i$ dan $E[R_d]_P$ berturut-turut merupakan nilai harapan dari *return* harian saham ke-i dan portofolio. Angka 252 merupakan jumlah hari perdagangan aset di pasar modal milik berbagai negara dalam setahun, khususnya Amerika Serikat. Sementara itu, standar deviasi tahunan portofolio atau suatu aset dapat dihitung menggunakan formula berikut (Bacon, 2021):

$$(\sigma_a)_i = (\sigma_d)_i \times \sqrt{252} \tag{28}$$

$$(\sigma_a)_P = (\sigma_d)_P \times \sqrt{252} \tag{29}$$

dengan $(\sigma_d)_i$ dan $(\sigma_d)_P$ merupakan standar deviasi dari imbal hasil harian saham ke-i dan portofolio.

1.6.7 Transformasi Data

Transformasi data merupakan pengubahan nilai-nilai suatu data dengan tujuan tertentu, misalnya memperbaiki persebaran data, menghilangkan *outlier* dan mengurangi *skewness*, atau memenuhi syarat-syarat analisis statistik (Hakimah et al., 2024). Hal ini dikarenakan beberapa metode analisis data memiliki sensitivitas tinggi pada inputannya, sehingga perlu dilakukan perubahan pada nilai data yang akan diinput. Secara umum, persebaran data hasil transformasi akan cenderung simetris atau bahkan menyerupai distribusi normal.

Pemilihan metode dan interpretasi hasil perlu diintegrasikan secara cermat jika ingin menghasilkan data yang optimal untuk analisis. Untuk memastikan keputusan metode transformasi data diambil secara bijaksana, peneliti diwajibkan untuk memahami secara mendalam metode transformasi yang tersedia dan implikasinya terhadap analisis statistik. Metode transformasi yang tepat dapat memperbaiki distribusi data sehingga memenuhi asumsi analisis dan mempertahankan informasi yang dimiliki oleh data awal (Hakimah et al., 2024). Transformasi data yang digunakan dalam penelitian ini akan terbatas pada *winsorizing*, *log-transformation*, dan *Yeo-Johnson transformation*.

1.6.7.1 *Winsorizing*

Data finansial seperti tingkat *return* dan rasio keuangan tidak jarang memiliki nilai pencilan atau *outlier*, yaitu nilai hasil pengamatan yang terletak jauh dari persebaran data lainnya. Keberadaan *outlier* pada data akan memberikan dampak yang signifikan pada hasil analisis (Ardhani et al., 2025). Dengan demikian, diperlukan suatu metode untuk menangani pencilan dalam data agar hasil analisis yang diperoleh dapat lebih akurat. Salah satunya metode tersebut adalah pendekatan *winsorizing*.

Winsorization merupakan metode penanganan pencilan dalam data dengan mengubah nilai-nilai ekstrem pada data sehingga mengurangi efek dari *outlier* (Wilcox, 2021). Umumnya, pendekatan ini dapat dilakukan dengan memodifikasi data ekstrem menjadi nilai persentil tertentu, seperti pada persentil ke-1 dan persentil ke-99, persentil ke-5 dan persentil ke-95, maupun persentil ke-10 dan persentil ke-90. Dengan demikian, data yang tergolong *outlier* diubah menjadi nilai yang lebih representatif serta analisis data hasil *winsorizing* dapat lebih presisi dan interpretatif (Ardhani et al., 2025).

1.6.7.2 *Log-Transformation*

Data yang memiliki *skewness* atau *outlier* dapat menjadi kendala dalam sejumlah pemodelan, salah satunya dalam model linear. Oleh karena itu, diperlukan transformasi yang dapat mengubah data mentah menjadi data baru dengan persebaran yang cenderung simetris atau mendekati distribusi normal. Salah satu metode yang paling umum digunakan adalah *log-transformation* (West, 2021).

Log-transformation merupakan teknik pengubahan nilai-nilai pada data dengan menggantinya menjadi hasil logaritma dari data tersebut. Basis yang digunakan untuk logaritmanya dapat berupa bilangan Euler (e) atau 10. Meskipun bilangan Euler memiliki peminat paling banyak, logaritma basis 10 berguna dalam mengubah data dengan nilai ratusan maupun ribuan menjadi data baru dengan nilai satuan (West, 2021). Dalam penelitian ini, bilangan Euler (e) digunakan sebagai basis dari transformasi logaritma.

1.6.7.3 *Yeo-Johnson Transformation*

Teknik transformasi data seperti menggunakan fungsi logaritma memiliki beberapa kelemahan. Salah satu kelemahan utamanya adalah ketidakmampuan dalam mentransformasikan data-data yang memiliki nilai negatif. Salah satu bentuk modifikasi dari transformasi logaritmik adalah transformasi Yeo-Johnson yang didefinisikan sebagai berikut (Yeo dan Johnson, 2000):

$$G_\theta(x) = \begin{cases} \frac{(1+x)^\theta - 1}{\theta} & , jika\ \theta \neq 0\ dan\ x \geq 0 \\ \ln(x+1) & , jika\ \theta = 0\ dan\ x \geq 0 \\ \frac{-[(-x+1)^{2-\theta} - 1]}{2-\theta} & , jika\ \theta \neq 2\ dan\ x < 0 \\ -\ln(-x+1) & , jika\ \theta = 2\ dan\ x < 0 \end{cases} \tag{30}$$

Perhitungan nilai parameter θ umumnya dilakukan menggunakan estimasi *likelihood* maksimum (MLE). Teknik estimasi ini rentan dipengaruhi oleh data pencilan. Bahkan satu data pencilan saja mampu memberikan pengaruh yang signifikan terhadap hasil perhitungan θ. Hal ini menjadikan hasil transformasi sangat sensitif terhadap *outlier* pada data (Nwakuya dan Anyaogu, 2022).

1.6.8 Analisis Komponen Utama

Analisis Komponen Utama atau *Principal Component Analysis* (PCA) merupakan metode statistik yang bertujuan untuk mengurangi dimensi dari data yang besar. Metode ini mengubah variabel asli yang saling berkorelasi satu sama lain menjadi satu set variabel yang lebih sedikit, tetapi masih mampu mencakup sebagian besar informasi yang dimiliki data asli. Secara umum, PCA digunakan untuk menganalisis struktur data yang kompleks serta hubungan antar variabel sehingga mudah dipahami (Lesnussa et al., 2025).

Langkah-langkah yang dilakukan untuk menentukan komponen utama adalah sebagai berikut (Rosyada dan Utari, 2024):

1. Standardisasi data pada setiap variabel agar memiliki skala nilai yang setara. Metode yang paling umum adalah dengan menggunakan *z-score*.
2. Buat matriks kovariansi ($\boldsymbol{\Sigma}$) untuk menentukan korelasi setiap variabel
3. Hitung nilai *eigen* dari matriks kovariansi menggunakan persamaan
$$|\lambda \boldsymbol{I} - \boldsymbol{\Sigma}| = 0 \quad (31)$$
4. Hitung vektor *eigen* yang berkorespondensi dengan nilai *eigen* yang telah diperoleh menggunakan Persamaan (4)
5. Tentukan komponen-komponen utama, yakni vektor $\mathbf{x}$, yang akan diambil dengan ketentuan nilai *eigen* yang berkorespondensi dengan vektor $\mathbf{x}$ lebih dari atau sama dengan 1.
6. Hitung transformasi data baru menggunakan persamaan
$$\boldsymbol{PC}_j = \boldsymbol{Z}\mathbf{x}_j \quad (32)$$
dengan $\boldsymbol{Z}$ merupakan matriks data hasil standardisasi dan $\mathbf{x}_j$ adalah vektor komponen ke-j.

Kegunaan PCA dalam mereduksi jumlah dimensi data berukuran besar sangat membantu dalam proses pengelompokan data. Hal ini dikarenakan kecepatan proses pengelompokan data sangat dipengaruhi oleh jumlah fitur data yang dimiliki. Selain itu, data hasil PCA umumnya hanya memiliki dua sampai tiga fitur saja, sehingga dapat divisualisasikan ke dalam dua atau tiga dimensi, yang memudahkan para peneliti untuk menganalisis struktur atau pola yang dimiliki data (Ding dan He, 2004).

Principal Component Analysis telah menjadi salah satu metode pengurangan dimensi data yang paling banyak digunakan dan umumnya diaplikasikan pada proses identifikasi, klasifikasi, dan kompresi gambar. Walaupun begitu, metode ini kurang efektif pada data yang memiliki pencilan atau *outliers*. Hal ini dikarenakan metode PCA menggunakan matriks variansi-kovariansi yang sangat sensitif terhadap *outliers*. Bahkan satu data *outlier* saja cukup untuk memberikan dampak negatif terhadap hasil PCA (Alkan et al., 2015).

1.6.9 Algoritma *Clustering*

Clustering adalah metode ekstraksi pengetahuan mengenai struktur tersembunyi dalam data dengan mempartisinya menjadi beberapa kelompok (*cluster*) tanpa menggunakan pada pengetahuan awal mengenai data tersebut. Hal yang mendasari penggunaan algoritma ini adalah untuk membandingkan titik-titik data. Data dibandingkan berdasarkan posisi, karakteristik ataupun perilakunya. Dalam pengambilan keputusan bisnis dan finansial, algoritma *clustering* berguna untuk mengelompokkan pelanggan sehingga dapat mengetahui target pasar, atau mengelompokkan aset-aset investasi berdasarkan kinerjanya untuk membentuk portofolio investor (Karlina dan Ario, 2021).

Clustering memiliki dua pendekatan utama yang sangat sering digunakan: pendekatan partisi (*flat*) dan pendekatan bertingkat (*hierarchical*). Pendekatan secara *flat* merupakan teknik pengelompokan data dengan cara memilah-milah data yang ada sebelum dimasukkan ke dalam suatu klaster. Sementara pendekatan secara *hierarchical* adalah teknik mengelompokkan data dengan membuat suatu hierarki berdasarkan pohon bertingkat. Data dengan tingkat kesamaan yang tinggi akan dimasukkan dalam hierarki yang berdekatan, sedangkan data yang memiliki tingkat kesamaan rendah akan ditempatkan dalam klaster yang berjauhan (Karlina dan Ario, 2021). Selain itu, terdapat pula dua pendekatan lainnya: pendekatan titik pusat (*centroid*) dan pendekatan kepadatan (*density*). Perbedaan utamanya terletak pada peletakan titik pusat setiap klasternya. Pendekatan secara *density* menetapkan pusat suatu klaster pada pusat kumpulan data, sehingga titik-titik data yang berada jauh dari kumpulan data yang padat dianggap sebagai *outlier* atau *noise*. Sementara itu, pendekatan *centroid* berfokus pada pemilihan titik pusat klaster sedemikan sehingga semua titik data yang ada di sekitarnya memiliki jarak yang hampir sama terhadap titik pusat (Xu dan Wunsch, 2009).

1.6.9.1 *K-Means*

Algoritma *K-Means* merupakan metode *clustering* paling umum digunakan karena tingkat efisiensinya yang tinggi dan dapat diterapkan pada hampir seluruh jenis data (Palupi et al., 2019). Algoritma ini bertujuan untuk mengelompokkan titik data ke dalam kelas-kelas yang memiliki variansi yang sama, atau lebih tepatnya, meminimalkan jumlah kuadrat dalam klaster. Oleh karena itu, algoritma ini sering direkomendasikan untuk distribusi data berbentuk bola (Thaidet dan Kawiwat A., 2022). Akan tetapi, metode ini sangat sensitif terhadap penempatan awal pusat data dan dapat dipengaruhi oleh data *outlier* (Jearanaitanakij et al., 2024).

Xu dan Wunsch, dalam bukunya yang berjudul *Clustering*, menjelaskan langkah-langkah algoritma *K-means* sebagai berikut (Xu dan Wunsch, 2009):

1. Inisiasi K sentroid klaster secara acak ataupun berdasarkan informasi yang telah diberikan
2. Memasukkan titik data ke dalam klaster yang memiliki sentroid terdekat. Pengukuran jarak antara titik data dengan sentroid menggunakan metrik jarak yang telah ditentukan, yaitu *euclidean*
3. Menentukan sentroid baru berdasarkan hasil pengelompokan sebelumnya memakai rumus berikut:

$$c_i = \frac{1}{N_{C_i}} \sum_{p_i \in C_i} p_i \tag{33}$$

4. Mengulangi langkah kedua dan ketiga sehingga tidak terjadi perubahan pada klaster yang terbentuk. Dengan kata lain, semua sentroid yang baru bernilai sama dengan sentroid dari iterasi sebelumnya.

1.6.9.2 *Agglomerative Hierarchy*

Agglomerative Hierarchical Clustering merupakan salah satu metode pengelompokan data yang menggunakan pendekatan dari bawah ke atas (*bottom-up*) dalam pembentukan klasternya. Hasil dari algoritma ini berupa sebuah dendrogram dengan setiap titik data berada pada level terendah (Zainuddin et al., 2024). Jarak antar klaster dapat diukur menggunakan sejumlah pendekatan seperti jarak minimum (*single linkage*), jarak rata-rata (*average linkage*), maupun jarak maksimum (*complete linkage*). Algoritma pengelompokan ini sangat berguna untuk mengungkap struktur hierarkis data, sehingga sering digunakan pada bidang bioinformatika dan analisis jaringan sosial (Jearanaitanakij et al., 2024). Dalam skripsi ini, peneliti menggunakan pendekatan jarak rata-rata (*average linkage*) untuk pembentukan klasternya.

Johnson dan Wichern, dalam bukunya yang berjudul *Applied Multivariate Statistical Analysis*, memaparkan tahapan klaster hierarki *agglomerative* untuk mengelompokkan n titik data, yaitu sebagai berikut (Johnson dan Wichern, 2007):

1. Konstruksikan n klaster dengan setiap klaster berisi satu titik data, dan sebuah matriks simetris $n \times n$ dari jarak (kemiripan) $\boldsymbol{D} = (d_{ij})$
2. Cari pasangan klaster dengan jarak paling dekat. Misalkan klaster yang dimaksud adalah C_i dan C_j, dengan jaraknya dinotasikan $d_{C_i C_j}$
3. Gabungkan klaster C_i dan C_j, kemudian beri label pada klaster baru dengan $(C_i C_j)$. Perbarui setiap entri pada matriks jarak dengan cara: (a) menghapus baris-baris dan kolom-kolom yang bersesuaian dengan klaster C_i dan C_j; dan (b) menambahkan sebuah baris dan kolom yang berupa jarak-jarak antara klaster $(C_i C_j)$ dengan klaster-klaster yang tersisa.
4. Ulangi langkah 2 dan 3 sebanyak $n - 1$ kali. Dengan demikian, semua titik data akan berada dalam satu klaster besar setelah algoritma selesai. Beri identitas pada setiap klaster yang digabungkan dan catat level tempat penggabungan klaster terjadi.

Pendekatan jarak rata-rata (*average linkage*) antara dua klaster dilakukan dengan menghitung jarak rata-rata semua titik data dalam satu klaster dengan seluruh titik data pada klaster lain. Setelah matriks jarak $\boldsymbol{D} = (d_{ij})$ terbentuk, akan dicari dua titik data dengan jarak terdekat. Misalkan C_i dan C_j adalah dua objek yang akan dikelompokkan sehingga membentuk klaster $(C_i C_j)$. Selanjutnya, perhitungan jarak klaster $(C_i C_j)$ dengan klaster C_k ataupun klaster lainnya dapat menggunakan rumus berikut:

$$d_{(C_i C_j) C_k} = \frac{\sum_{p_i \in (C_i C_j)} \sum_{p_k \in C_k} d_{ik}}{N_{(C_i C_j)} N_{C_k}} \tag{34}$$

dengan d_{ik} merupakan jarak antara titik data p_i dari klaster $(C_i C_j)$ dan titik data p_k pada klaster C_k, serta $N_{(c_i c_j)}$ dan N_{C_k} berturut-turut merupakan jumlah titik data dalam klaster $(C_i C_j)$ dan C_k.

1.6.9.3 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) merupakan metode *clustering* yang bekerja dengan cara memperluas suatu wilayah yang memiliki kepadatan signifikan menjadi sebuah klaster (Ahmad dan Dang, 2015). Algoritma DBSCAN mengasumsikan klaster sebagai himpunan maksimum dari titik-titik yang terhubung secara kepadatan. Titik data yang tidak termasuk dalam klaster dan akan dianggap sebagai *noise* (Sutramiani et al., 2024).

Jumlah klaster yang dihasilkan oleh algoritma DBSCAN dipengaruhi oleh dua parameter, yaitu epsilon (ε) dan titik minimum. Titik minimum atau MinPts merupakan jumlah titik minimum yang diperlukan untuk membentuk sebuah klaster. Nilai MinPts yang terlalu kecil akan membuat proses *clustering* sangat sensitif terhadap *noise*, dan jika terlalu besar maka akan menghasilkan klaster yang kurang fleksibel. Epsilon atau Eps merupakan jarak yang digunakan untuk menetapkan wilayah klaster. Apabila dua buah titik berjarak tidak melebihi nilai Eps, maka kedua titik tersebut berada dalam satu klaster. Nilai Eps yang terlalu besar mengakibatkan adanya penggabungan dua klaster yang semestinya berbeda saat proses pengelompokan sedang berjalan, sedangkan jika nilainya terlalu kecil maka akan menyebabkan pembentukan klaster-klaster kecil.

Berikut ini adalah tahapan algoritma DBSCAN secara garis besar:

1. Menentukan nilai MinPts dan Eps
2. Menghitung jarak antar titik data
3. Memasukkan titik data dengan jarak kurang dari sama dengan Eps ke dalam wilayah *neighborhood* yang sedang diobservasi
4. Melabeli setiap titik sebagai *core point* jika jumlah *neighborhood* lebih dari sama dengan MinPts. Apabila ada titik data dengan syarat *core point* yang tidak terpenuhi, maka titik tersebut dikategorikan sebagai *noise*
5. Masukkan semua titik data yang memiliki hubungan *neighborhood* dan bukan *noise* ke dalam satu klaster

1.6.9.4 *Silhouette Score*

Silhouette score adalah metrik yang berfungsi mengukur kualitas klaster dalam analisis *clustering*. Metrik ini merupakan hasil penggabungan metode separasi yang berfungsi untuk mengukur seberapa signifikan perbedaan antara suatu klaster dengan klaster lainnya dan metode kohesi yang mengukur kedekatan antar titik data dalam sebuah klaster (Rousseeuw, 1987). *Silhouette score* merupakan rata-

rata dari koefisien *silhouette* dari setiap titik data. Kauffman dan Rousseeuw mendefinisikan koefisien *silhouette* pada titik data ke-i, yakni p_i, sebagai berikut (Kauffman dan Rousseeuw, 2005):

$$Koefisien\ Silhouette_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \tag{35}$$

dengan a_i sebagai rata-rata jarak titik data ke-i dengan semua titik lainnya di dalam klaster dan b_i sebagai rata-rata terkecil jarak titik data ke-i ke klaster lainnya. Nilai dari a_i dan b_i berturut-turut dapat dihitung menggunakan formula berikut:

$$a_i = \frac{\sum_{p_i, p_j \in C_i} d_{ij}}{N_{C_i}}; b_i = \min_{C_k \neq C_i} \frac{\sum_{p_i \in C_i, p_k \in C_k} d_{ik}}{N_{C_k}}$$

a_i merupakan ukuran kemiripan data ke-i dengan data-data lainnya dalam klaster yang sama. Berdasarkan formula dari a_i, semakin kecil nilai a_i menandakan semakin tepat data ke-i ditempatkan dalam sebuah klaster. Sementara itu, b_i merupakan ukuran kemiripan data ke-i dengan data-data di klaster lain. Berdasarkan formula dari b_i, semakin besar nilai b_i menandakan semakin tidak cocok data ke-i ditempatkan di klaster lain (Kauffman dan Rousseeuw, 2005).

Silhouette score memiliki rentang nilai dari -1 hingga 1, dengan skor yang semakin tinggi menunjukkan pembagian titik-titik data yang semakin bagus. *Silhouette score* yang positif dan semakin mendekati 1 mengindikasikan klaster-klaster yang dibentuk terdefinisi dengan baik, skor yang semakin mendekati 0 menandakan klaster-klaster yang terbentuk saling tumpang tindih, serta *silhouette score* yang semakin mendekati -1 menunjukkan bahwa bisa jadi titik-titik data telah dikelompokkan ke dalam klaster yang salah (Jearanaitanakij et al.,2025).

1.6.10 Algoritma Metaheuristik

Metaheuristik merupakan gabungan dari dua kata Yunani, yakni *meta* yang berarti "metodologi tingkat atas" dan *heuriskein* yang berarti seni dalam menemukan strategi untuk memecahkan masalah. Metode metaheuristik dapat didefinisikan sebagai metodologi tingkat atas yang dapat digunakan sebagai strategi dalam pemecahan masalah optimasi spesifik (Talbi, 2009).

Keunggulan utama dari algoritma metaheuristik adalah fleksibilitas dan kemampuannya yang serbaguna. Hal ini dikarenakan algoritma ini dapat dimodifikasi untuk memenuhi persyaratan spesifik suatu masalah, menjadikannya solusi ideal untuk berbagai masalah optimasi di berbagai bidang teknik dan sains. Misalnya, metaheuristik telah diterapkan dengan baik dalam penjadwalan transportasi, dalam mendesain bangunan dan fasilitas umum, dan dalam *data mining* seperti klasifikasi, prediksi, *clustering*, dan pemodelan sistem (Tomar et al., 2023).

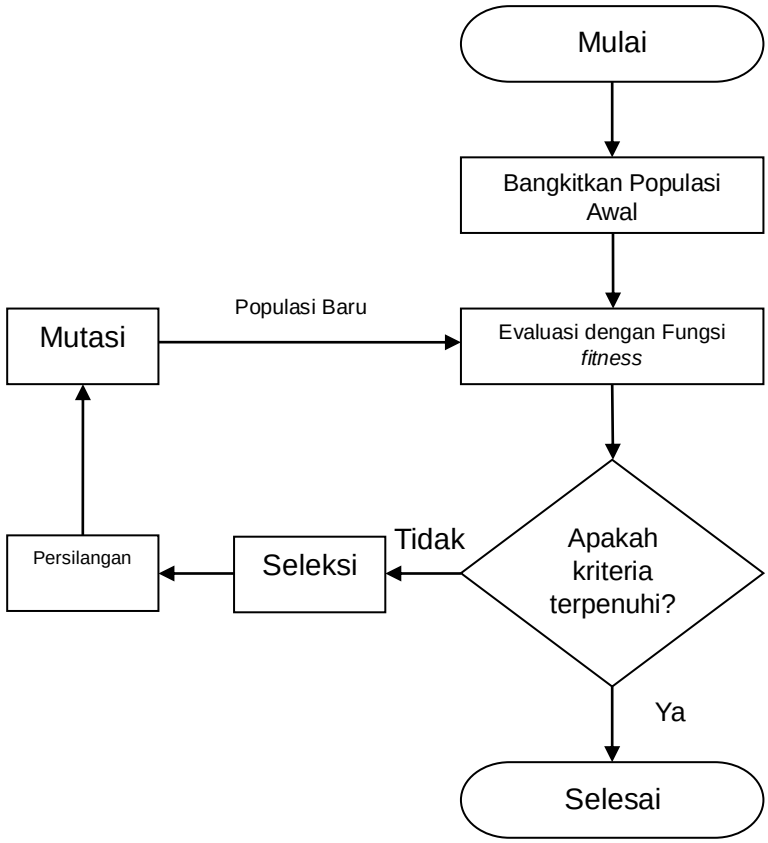
Terobosan metaheuristik dikelompokkan ke dalam empat kategori berdasarkan perilakunya: *human-based*, *physics-based*, *evolution-based*, dan *based on swarm intellects*. Keempat kelompok algoritma ini terinspirasi dari fenomena alam dan memiliki sejarah panjang, yang dapat ditelusuri hingga tahun-tahun awal perkembangannya. Keempat kelompok ini juga dinamakan algoritma *Nature-Inspired Intelligent* (NII) dan dikenal karena kemampuannya untuk menyempurnakan populasi kandidat menggunakan informasi yang diperoleh selama proses algoritma berlangsung (Tomar et al., 2023). Penelitian ini akan berfokus pada penggunaan algoritma genetik, *particle swarm optimization*, serta gabungannya.

1.6.10.1 Algoritma Genetik

Algoritma genetik merupakan algoritma metaheuristik yang termasuk dalam kategori *evolution-based*, yaitu algoritma yang alur kerjanya terinspirasi dari proses evolusi yang terjadi di alam (Hassanat et al., 2019). Algoritma ini pertama kali diperkenalkan oleh Holland di tahun 1975 sebagai metode pemecahan masalah optimasi, dengan memanfaatkan konsep genetik, seleksi alam, dan alur evolusi makhluk hidup (Holland, 1975). Sebagai metode optimasi, algoritma genetik telah dimanfaatkan dalam menyelesaikan masalah di berbagai bidang, seperti jaringan komputer, rekayasa perangkat lunak, pemrosesan gambar, pengenalan suara, kesehatan, permainan komputer, dan sebagainya (Hassanat et al., 2019).

Algoritma genetik adalah metode optimasi global yang dibentuk dengan mensimulasikan proses evolusi seleksi alam dan populasi "*survival of the fittest*". Setiap solusi yang memungkinkan diekspresikan sebagai "kromosom" sehingga didapat "kelompok" yang terdiri dari kromosom-kromosom tersebut. Kelompok ini dibatasi pada lingkungan dengan kondisi tertentu. Fungsi objektif digunakan untuk mengevaluasi kualitas setiap individu dalam lingkungan tersebut. Individu yang lebih adaptif terhadap lingkungan memiliki peluang bertahan hidup yang lebih tinggi. Sejumlah individu dimunculkan secara acak, lalu dikombinasikan melalui kawin silang berdasarkan prinsip "*survival of the fittest*" untuk menghasilkan keturunan yang lebih baik daripada generasi sebelumnya. Dengan cara ini, populasi "kromosom" akan terus berevolusi secara bertahap menjadi solusi yang lebih baik. Penggabungan operasi genetika seperti mutasi gen dalam proses evolusi akan meningkatkan kualitas keturunan menjadi lebih adaptif dengan kondisi lingkungan yang diberikan (Tang dan Wu, 2021).

Tahapan algoritma genetik secara umum disajikan secara ringkas dalam *flowchart* berikut:



Gambar 1.1 Alur kerja algoritma genetik

Algoritma genetik memiliki beberapa parameter yang mesti diinisialisasi dari awal sebelum menjalankannya, yakni sebagai berikut:

a. Ukuran populasi, yakni total individu yang dibangkitkan dalam populasi. Ukuran populasi yang kecil akan membuat ruang pencarian menjadi terbatas, sehingga dikhawatirkan hasil algoritma berupa nilai optimum lokal. Di sisi lain, populasi yang besar dapat memperluas ruang pencarian untuk solusi optimal, tetapi akan menambah beban komputasi ketika menjalankannya. Oleh karena itu, ukuran populasi mesti tergolong wajar (Roeva et al., 2013).
b. Peluang pindah silang, yaitu kemungkinan terjadinya persilangan kromosom dalam satu generasi. Nilai probabilitas sebesar 100% menandakan semua keturunan diperoleh melalui pindah silang, sedangkan nilai 0% mengindikasikan semua keturunan sama persis dengan generasi sebelumnya (De Jong dan Spears, 1992).
c. Peluang mutasi merupakan kemungkinan jumlah kromosom yang mesti dimutasikan dalam satu generasi. Mutasi bertujuan untuk mencegah algoritma genetik menghasilkan nilai optimum lokal. Akan tetapi, jika peluang

mutasi terlalu besar, maka algoritma genetik akan berubah menjadi pencarian acak (Lynch, 2010).

Berdasarkan alur kerja pada Gambar 1.1, algoritma genetik memiliki tiga operator, yaitu:

a. Seleksi merupakan tahap penentuan suatu individu akan mengikuti proses reproduksi atau tidak dalam algoritma genetik. Operator seleksi sering disebut sebagai operator reproduksi (Jebari dan Madiafi, 2013). Operator-operator seleksi yang umum digunakan adalah sebagai berikut (Katoch et al., 2020):
   a) Seleksi roda (*roulette wheel*) menempatkan seluruh individu ke dalam roda berputar dengan proporsi yang mengacu pada nilai fungsi *fitness*-nya. Roda ini kemudian diputar secara acak untuk memperoleh individu yang akan digunakan untuk menghasilkan keturunan baru.
   b) Seleksi *ranking* merupakan hasil modifikasi dari seleksi roda yang diperoleh dengan cara mengganti roda berputar menjadi sistem pemeringkatan. Individu dengan nilai *fitness* tertinggi akan berada di peringkat terbaik (M) dan peringkat terburuk (1) untuk individu dengan *fitness* terendah (Shukla et al., 2015).
   c) Seleksi turnamen pertama kali diperkenalkan oleh Brindle di tahun 1983. Sepasang individu dipilih berdasarkan nilai fungsi *fitness*-nya, lalu menyimpan individu dengan nilai *fitness* tertinggi untuk pembentukan generasi berikutnya. Proses ini dilakukan secara terus-menerus sampai setiap individu dibandingkan dengan $M - 1$ individu lainnya (Katoch et al., 2020).
b. Persilangan merupakan proses penyeleksian titik-titik di kromosom dari orang tua yang akan disilangkan sehingga menghasilkan keturunan baru. Beberapa teknik persilangan yang umum digunakan adalah sebagai berikut (Lim et al., 2017):
   a) Persilangan satu titik (*single point crossover*), yaitu metode persilangan dengan memilih satu titik di kromosom secara acak, lalu membagi kromosom masing-masing individu menjadi dua segmen pada titik yang sudah dipilih sebelum akhirnya kedua individu saling bertukar segmen dan menghasilkan keturunan baru. Bentuk umum dari teknik ini adalah persilangan K-titik (K-*point crossover*), yakni memilih K titik secara acak dan membagi kromosom masing-masing individu menjadi $K + 1$ segmen, lalu kedua individu saling bertukar segmen sehingga dihasilkan keturunan baru (Lim et al., 2017).
   b) Persilangan seragam tidak membagi kromosom ke dalam beberapa segmen untuk pembuatan keturunan. Keturunan hasil persilangan diperoleh dengan menyalin kromosom orang tua dengan proporsinya masing-masing. Oleh karena itu, keturunan yang diperoleh merupakan campuran dari kromosom kedua orang tuanya. Akan tetapi, terdapat

kemungkinan keturunan yang dihasilkan akan sama persis dengan salah satu dari orang tuanya (Kora dan Yadlapalli, 2017).

Akan tetapi, penelitian ini tidak akan menggunakan bilangan biner 1 dan 0 dalam pengkodean nilai untuk kromosomnya, melainkan vektor. Hal ini dikarenakan untuk penggabungan GA dan PSO, penggunaan vektor menghasilkan performa yang lebih baik dibandingkan dengan penggunaan bilangan biner (Kao dan Zahara, 2007).

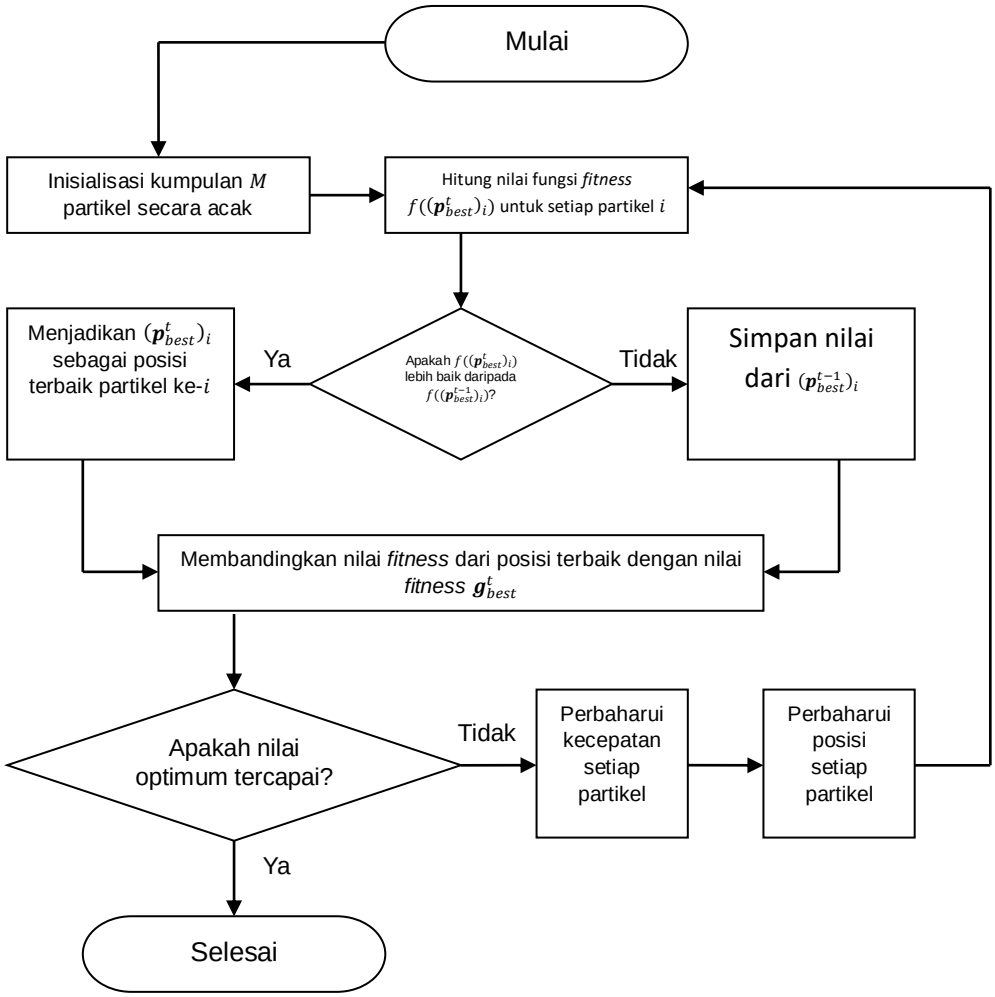
c. Mutasi merupakan operator yang mengubah satu atau beberapa gen dari kromosom keturunan tepat setelah persilangan dilakukan. Mutasi dilakukan dengan harapan dapat memperkecil peluang diperolehnya solusi optimum lokal dan memperkaya keanekaragaman individu dalam populasi (Hassanat et al., 2019). Teknik-teknik mutasi yang umum digunakan adalah sebagai berikut (Katoch et al., 2020):
   a) *Displacement mutation* merupakan operasi memindahkan segmen kromosom dalam individu. Posisi perpindahan segmen dipilih secara acak sehingga hasil mutasi yang diperoleh valid.
   b) *Simple inversion mutation* adalah operasi pembalikan dua segmen kromosom individu yang dipilih secara acak dan meletakkannya di posisi yang acak pula.
   c) *Scramble mutation* menempatkan gen-gen di segmen kromosom individu dengan panjang tertentu ke dalam posisi yang acak. Operator ini diharapkan dapat meningkatkan nilai *fitness* dari individu tersebut.

1.6.10.2 *Particle Swarm Optimization*

Optimasi kawanan partikel atau *particle swarm optimization* (PSO) pertama kali diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995 dan merupakan algoritma optimasi yang cara kerjanya terinspirasi dari perilaku sosial hewan yang hidup bergerombol, misalnya burung dan ikan (Kennedy dan Eberhart, 1995). Perilaku hewan tersebut diterjemahkan ke dalam operasi algoritmik untuk menyelesaikan masalah optimasi dengan cara menginterpretasikan gerombolan hewan sebagai kumpulan partikel dan setiap partikel mewakili kandidat solusi. Gerombolan partikel mengeksplorasi ruang dalam dimensi yang diberikan dan menemukan solusi terbaik untuk masalah yang dihadapi (Shami et al., 2022).

Algoritma PSO dikenal karena kemudahan dalam efisiensi dan implementasinya yang hanya membutuhkan beberapa baris kode. Algoritma ini telah diterapkan ke dalam bidang teknik, penyaringan data, *machine learning*, dan bidang lainnya (Thaher et al., 2022; Houssein dan Sayed, 2023). Keunggulan lain dari algoritma PSO adalah mudah digabungkan dengan algoritma optimasi lainnya, misalnya algoritma genetik dan *differential evolution* (Shami et al., 2022).

Algoritma PSO dimulai dengan membangkitkan sebuah kumpulan partikel acak terlebih dahulu. Setiap partikel memiliki posisi dan kecepatannya masing-masing dan akan mengingat posisi terbaiknya atau bisa dinotasikan $(\boldsymbol{p}_{best}^{t})_i$, sedangkan posisi terbaik partikel lainnya dari seluruh kawanan dinotasikan $\boldsymbol{g}_{best}^{t}$. Setiap partikel akan bergerak di ruang pencarian menuju posisi terbaik berdasarkan nilai fungsi *fitness* dan menuju tempat partikel terbaik berada (Kennedy dan Eberhart, 1995). Proses kerja PSO ditampilkan dalam bentuk *flowchart* sebagai berikut (Gad, 2022):



Gambar 1.2 Alur kerja algoritma *particle swarm optimization*

Algoritma PSO membutuhkan sejumlah parameter yang mesti ditetapkan terlebih dahulu sebelum dijalankan, yakni sebagai berikut (Wang et al., 2017; Abualigah, 2025):

a. Jumlah partikel merupakan total partikel yang diperlukan untuk mencari solusi paling optimal. Jumlah partikel yang kecil mengakibatkan ruang

pencarian kawanan menjadi berkurang sehingga menghasilkan solusi yang tidak memuaskan. Di sisi lain, jumlah partikel yang besar dapat meningkatkan kualitas dari solusi yang diperoleh, tetapi akan meningkatkan kompleksitas komputasi. Jumlah partikel yang dibangkitkan juga bergantung pada fungsi *fitness* yang akan dioptimalkan (Van den Bergh dan Engelbrecht, 2001). Jumlah partikel yang umum digunakan dalam algoritma PSO berkisar dari 20 hingga 50 partikel (Piotrowski et al., 2020).

b. Beban inersia adalah parameter yang menentukan besar pengaruh gerakan partikel sebelumnya terhadap gerakan saat ini (Abualigah, 2025). Beban inersia digunakan untuk menyeimbangkan pencarian solusi optimum global dan lokal (Wang et al., 2017). Nilai beban inersia disarankan berada dalam interval [0.8, 1.2] (Shi dan Eberhart, 1998).
c. Kecepatan maksimum atau V_{max} diperlukan sebagai batasan untuk mengendalikan kemampuan partikel dalam mencari nilai optimum global (Wang et al., 2017). Meskipun begitu, menentukan nilai V_{max} tidak mudah dan akan mengganggu kinerja PSO jika tidak dipilih dengan tepat. Nilai V_{max} yang besar mengakibatkan gerak partikel berpotensi terlalu cepat dan melewatkan solusi optimal. Sementara itu, nilai V_{max} yang kecil berpotensi mengurangi cakupan pencarian dari partikel, sehingga hanya menghasilkan solusi optimum lokal (Shami et al., 2022). Secara umum, V_{max} ditetapkan pada rentang yang bergantung pada jumlah variabel data (Wang et al., 2017).
d. Posisi maksimum atau X_{max} merupakan batas posisi partikel sehingga mencegahnya keluar dari ruang pencarian. Penentuan nilai X_{max} dipengaruhi oleh jumlah dimensi data yang diberikan dan posisi dari solusi optimum global terhadap optimum lokal (Wang et al., 2017).
e. Koefisien percepatan terdiri atas c_1 dan c_2 yang berfungsi untuk mengarahkan pencarian partikel menuju solusi optimal. Nilai c_1 yang lebih tinggi daripada c_2 akan mengubah PSO menjadi pencarian acak, sedangkan jika c_1 yang lebih kecil daripada c_2 akan menurunkan kualitas solusi optimal yang dihasilkan (Kennedy dan Eberhart, 1995). Salah satu penetapan koefisien percepatan yang umum digunakan adalah $c_1 = c_2 = 2.0$ (Wang et al., 2017).
f. Koefisien acak terdiri atas r_1 dan r_2 berasal dari distribusi seragam [0,1] dan berfungsi untuk memastikan variabilitas dalam proses pencarian sehingga mencegah diperolehnya solusi yang kurang optimal dan memperluas cakupan eksplorasi di ruang pencarian (Abualigah, 2025).

1.6.10.3 Algoritma Hibrida

Belakangan ini, popularitas algoritma hibrida meningkat dalam menangani masalah optimasi. Banyak algoritma metaheuristik dibuat dan dikembangkan dengan menggabungkan operator-operator paling efektif dari algoritma metaheuristik yang sudah ada. Penggabungan ini membantu penghindaran jebakan optimasi lokal untuk menjelajahi ruang pencarian secara efisien dan efektif, serta mencapai

penggunaan yang lebih baik. Dengan demikian, algoritma yang dibentuk memberikan solusi ideal dengan kualitas yang lebih tinggi dan tingkat efisiensi yang lebih besar (Tomar et al., 2023).

Genetic Algorithm-Particle Swarm Optimization (GA-PSO) merupakan salah satu algoritma metaheuristik gabungan yang menggabungkan alur kerja algoritma genetik dan *particle swarm optimzation*. Inti dari algoritma ini adalah mencari solusi optimal menggunakan kumpulan partikel yang dipandang sebagai populasi. Sebagian dari partikel yang ada akan dipersilangkan dan/atau dimutasi gennya pada setiap iterasinya sehingga menghasilkan populasi baru yang akan memberikan solusi terbaik. Langkah kerja dari algoritma ini adalah sebagai berikut (Kuo dan Hong, 2013):

1. Menentukan parameter algoritma, yang terdiri dari jumlah partikel, tingkat mutasi, tingkat persilangan, bobot inersia, ukuran dimensi, kecepatan maksimum, iterasi maksimum, posisi maksimum, posisi minimum, dan koefisien percepatan. Jumlah populasi atau partikel yang dibangkitkan dalam penelitian ini adalah sebanyak 40 individu. Pada kasus optimasi portofolio, setiap partikel memiliki bentuk vektor sebagai berikut:
$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}; w_i \in [0,1]$$
dan seluruh entri pada vektor tersebut memenuhi Persamaan (19).
2. Mengevaluasi kinerja partikel menggunakan fungsi *fitness*, lalu mengurutkannya dari terbesar ke terkecil berdasarkan nilainya sebagai bentuk seleksi. Fungsi *fitness* yang digunakan disesuaikan dengan kondisi portofolio yang optimal—imbal hasil maksimum sekaligus risiko yang minimum. Fungsi *fitness* yang akan digunakan untuk optimasi portofolio saham didefinisikan sebagai berikut:
$$Max(S_P) = Max\left(\frac{E[R_a]_P - r_f}{(\sigma_a)_P}\right) \quad (36)$$
dengan,
S_P : fungsi *fitness*
$E[R_a]_P$: nilai harapan tingkat *return* tahunan portofolio
r_f : tingkat *return* tahunan dari aset bebas risiko
$(\sigma_a)_P$: standar deviasi *return* tahunan portofolio
3. Mengsubtitusi setengah partikel dengan nilai *fitness* terkecil dengan setengah partikel dengan nilai *fitness* terbesar, sehingga ½*M* teratas sama dengan ½*M* terbawah
4. Menyimpan ½*M* partikel teratas dan membaharui ½M partikel terbawah. Pembaharuan partikel dilakukan dengan memakai formula berikut:
$$\boldsymbol{v}_i^{t+1} = \omega.\boldsymbol{v}_i^t + c_1.rand_1.((\boldsymbol{p}_{best}^t)_i - \mathbf{x}_i^t) + c_2.rand_2.(\boldsymbol{g}_{best}^t - \mathbf{x}_i^t) \quad (37)$$
$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \boldsymbol{v}_i^{t+1} \quad (38)$$

dengan,

ω : bobot inersia
$\boldsymbol{v}_i^t$: kecepatan partikel ke-i sekarang
$\boldsymbol{v}_i^{t+1}$: kecepatan partikel ke-i selanjutnya
c_1, c_2 : koefisien percepatan
$(\boldsymbol{p}_{best}^t)_i$: posisi terbaik partikel ke-i sekarang
$\boldsymbol{g}_{best}^t$: posisi terbaik semua partikel sekarang
$\mathbf{x}_i^t$: posisi partikel ke-i sekarang
$\mathbf{x}_i^{t+1}$: posisi partikel ke-i selanjutnya
$rand_1, rand_2$: bilangan acak yang berdistribusi seragam $[0,1]$

5. Pembaharuan dilakukan memakai rumus PSO, dilanjutkan dengan persilangan, dan diakhiri dengan mutasi. Dalam penelitian ini, tingkat persilangan yang digunakan adalah 60%, dengan formula persilangan yang didefinisikan sebagai berikut:

$$\mathbf{x}_i^{new} = \mathbf{x}_i^{old}.rand_3 + \mathbf{x}_{i+1}^{old}.(1 - rand_3), i = 1,2, \dots, \frac{1}{2}M - 1$$

$$\mathbf{x}_i^{new} = \mathbf{x}_i^{old}.rand_3 + \mathbf{x}_1^{old}.(1 - rand_3), i = \frac{1}{2}M \quad (39)$$

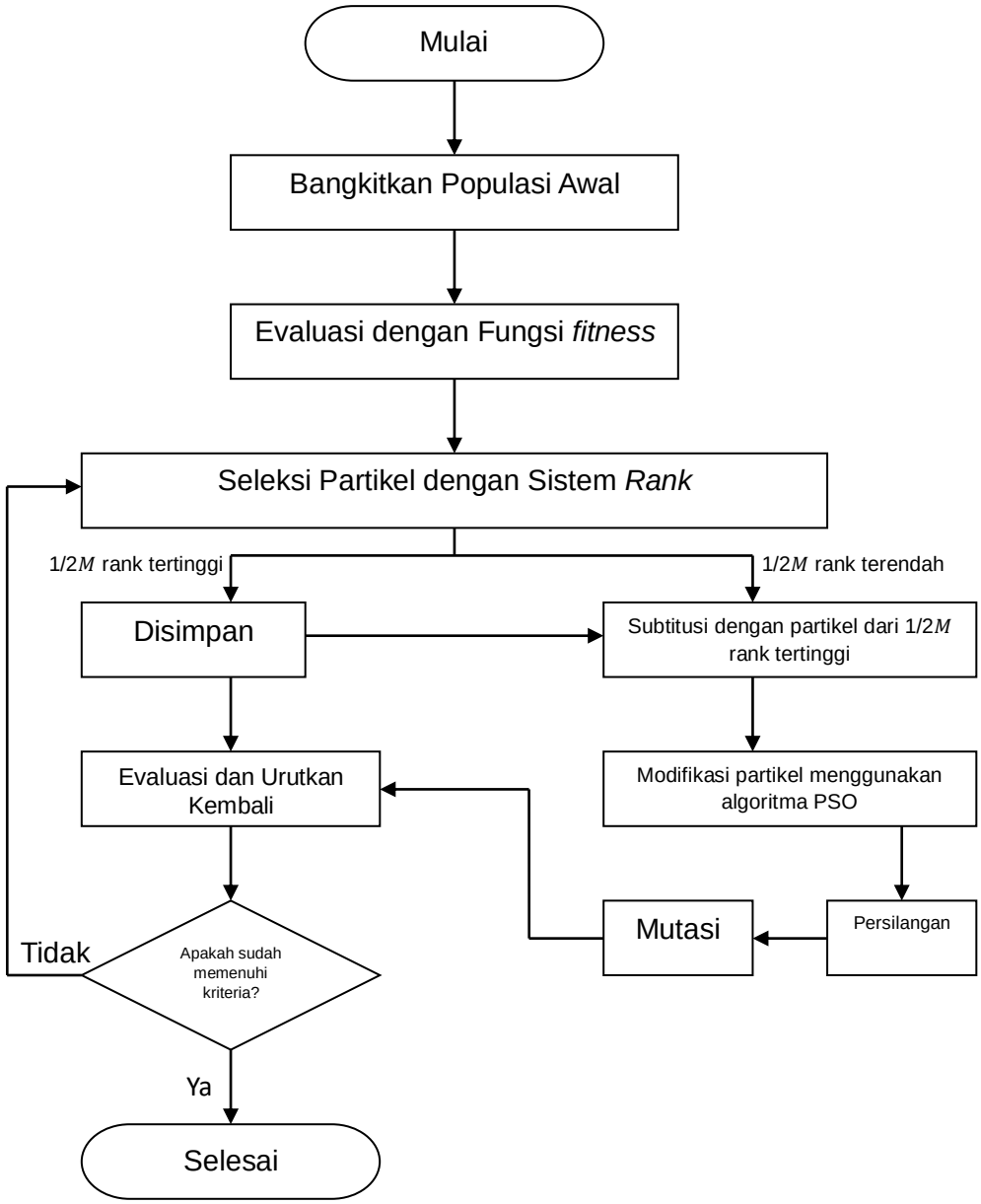
dengan $rand_3$ berasal dari distribusi seragam $[0,1]$.

Sementara itu, tingkat mutasi yang digunakan sebesar 20% dengan formula yang didefinisikan sebagai berikut:

$$\mathbf{x}_i^{mut} = \mathbf{x}_i^{new} + rand_4.\boldsymbol{N}(\mathbf{0}, \mathbf{1}) \quad (40)$$

dengan $rand_4$ berasal dari distribusi seragam $[0,1]$ dan $\boldsymbol{N}(\mathbf{0}, \mathbf{1})$ adalah vektor dengan entri bilangan acak yang berdistribusi normal standar

6. Mengevaluasi kembali keseluruhan partikel menggunakan fungsi *fitness*
7. Apabila sudah sesuai kriteria, algoritma berhenti, jika belum, kembali ke langkah ke-3.



Gambar 1.3 Alur kerja algoritma GA-PSO

1.7 Penelitian Terdahulu

Tinjauan terhadap literatur akademis di luar maupun dalam Indonesia menunjukkan bahwa komponen utama dalam penelitian ini—yaitu algoritma *clustering* dan optimasi metaheuristik—telah menjadi subjek penelitian yang cukup aktif. Sintesis dari penelitian-penelitian terdahulu membantu memetakan lanskap yang ada dan mengidentifikasi celah penelitian (*research gap*) yang diisi oleh studi ini.

Pertama, penerapan algoritma *clustering* telah terbukti efektif dalam membantu pengelompokan saham untuk pembuatan portofolio. Penelitian Nisardi menerapkan algoritma *clustering K-Means* menggunakan fitur *expected return* dan *volatilitas* untuk memilih rekomendasi saham yang akan dimasukkan ke dalam portofolio. Berdasarkan pilihan saham-saham yang diberikan, portofolio *Minimum Variance* dan *Tangency Portfolio* menjadi model Markowitz terbaik (Nisardi et al., 2025). Selain itu, penelitian yang dilakukan oleh Bulani menunjukkan bahwa rekomendasi saham hasil algoritma *clustering K-Medoids* memberikan kinerja portofolio yang lebih baik ketimbang hanya menggunakan saham-saham dengan kapitalisasi pasar tertinggi (Bulani et al., 2025).

Kedua, terdapat banyak algoritma *clustering* yang dapat digunakan dalam mengelompokkan saham-saham. Oleh karena itu, perlu dilakukan seleksi algoritma menggunakan sejumlah kriteria. Studi kasus Jearanaitanakij dan koleganya berhasil menyeleksi sejumlah algoritma *clustering* yang umum digunakan. Hasil penelitiannya menunjukkan algoritma *Gaussian Mixture* secara konsisten mengelompokkan saham indeks SET50 dengan baik karena memiliki rata-rata dari *Silhouette Score, Calinski-Harabasz Indeks,* dan *Davies-Bouldin Indeks* tertinggi (Jearanaitanakij et al., 2025) Perbandingan algoritma *clustering* juga dilakukan oleh Sutramiani beserta rekan-rekannya yang membandingkan performa *K-Means* dan DBSCAN. Hasil penelitiannya menunjukkan bahwa algoritma *K-Means* memiliki performa yang lebih bagus dalam mengelompokkan data UMKM di wilayah Aceh (Sutramiani et al., 2024).

Ketiga, pengembangan algoritma metaheuristik seperti *genetics algorithm* dan *particle swarm optimization* telah beberapa kali dilakukan. Anam dan rekan-rekannya menunjukkan bahwa kinerja portofolio buatan algoritma PSO dengan operator mutasi lebih bagus daripada algoritma PSO klasik, terlebih-lebih portofolio model Markowitz (Anam et al., 2025). Selain itu, penelitian yang dilakukan oleh Kuo dan Hong berhasil menggabungkan algoritma GA dan PSO dalam pembobotan reksadana (*mutual funds*) dari Taiwan. Penelitian tersebut juga menunjukkan bahwa kinerja portofolio buatan algoritma gabungan lebih bagus daripada algoritma GA maupun algoritma PSO, terlebih lagi kinerja pasar modal (Kuo dan Hong, 2013).

Meskipun topik-topik ini telah diteliti, penelitian ini menempatkan kontribusinya secara unik. Sebagian besar penelitian cenderung mengelompokkan saham tanpa melakukan metode optimasi pada pembuatan portofolionya. Sebagian penelitian lainnya berfokus pada optimasi pada pembobotan aset untuk portofolio tanpa menyeleksi aset yang digunakan algoritma *clustering* terlebih dahulu. Apabila terdapat keduanya, penelitian tersebut lebih menitikberatkan pada kualitas optimasi pembobotan aset dan menggunakan satu algoritma *clustering* untuk pengelompokan aset untuk portofolionya. Salah satu contohnya adalah penelitian pada saham LQ-45 menggunakan *self-organizing map* untuk pengelompokan saham dan algoritma genetik sebagai metode pembobotan saham perwakilan setiap klaster (Farid dan Rosadi, 2022).

Berikut adalah ringkasan dari penelitian-penelitian terdahulu yang relevan.

Tabel 1.1 Ringkasan penelitian terdahulu

Peneliti (Tahun)	Fokus Utama Penelitian	Metodologi Kunci	Temuan Relevan / Kontribusi
Anam, S. et al. (2025)	Pembentukan portofolio *Mean-VaR* pada saham perbankan	PSO dengan operator mutasi, *Mean-VaR*	Portofolio yang dibuat memiliki kinerja yang lebih baik dibandingkan buatan PSO klasik
Sutramiani, N. P. et al. (2024)	Perbandingan algoritma *clustering* mengelompokkan data UMKM	*K-Means*, DBSCAN, DBI *Score*	Algoritma *K-Means* membagi data menjadi beberapa klaster dengan kualitas yang lebih baik daripada DBSCAN.
Bulani, V. et al. (2025)	Pembentukan portofolio optimal pada saham Nasdaq100 dan *cryptocurrency*	*K-Medoids*, *Standard* PSO, *Improved* PSO, *Drift* PSO	Portofolio yang dibuat berdasarkan analisis klaster memberikan kinerja yang lebih bagus daripada hanya berfokus pada aset dengan kapitalisasi pasar yang tinggi
Jearanaitanakij, K. et al. (2025)	Membandingkan hasil klaster saham indeks SET50 dari tahun ke tahun berdasarkan kinerja perusahaan	*K-Means*, *Affinity Propagation*, *Agglomerative Hierarchy*, *Gaussian Mixture*, HDBSCAN	Menggunakan rataan dari *Silhouette Score, Calinski-Harabasz Indeks,* dan *Davies-Bouldin Indeks*, didapat *Gaussian Mixture* sebagai model terbaik

Peneliti (Tahun)	Fokus Utama Penelitian	Metodologi Kunci	Temuan Relevan / Kontribusi
Nisardi, M. K. et al. (2025)	Membentuk portofolio optimal dari saham indeks IDX-MES BUMN17 berdasarkan hasil klaster	*K-Means*, Model Markowitz,	Menggunakan rataan dan standar deviasi *return* sebagai fitur, dibentuk enam klaster berbeda, model Markowitz yang direkomendasikan adalah *Minimum Variance Portfolio* dan *Tangency Portfolio*
Farid, F. dan D. Rosadi (2022)	Membentuk portofolio optimal dari saham LQ-45 berdasarkan hasil *clustering*	*Self-Organizing Map*, Algoritma Genetik, *Mean-Variance*	Menggunakan rataan dan risiko dari *return* sebagai fitur, dibentuk tiga klaster berbeda, portofolio optimal buatan algoritma genetik menghasilkan kinerja yang lebih baik daripada *mean-variance*
Kuo, R. J. dan C. W. Hong (2013)	Menyeleksi *mutual fund* Taiwan untuk membentuk portofolio optimal	DEA, GA-PSO	Portofolio buatan algoritma gabungan memiliki kinerja yang lebih bagus daripada GA klasik, PSO klasik, dan Indeks Pasar

BAB II

METODE PENELITIAN

2.1 Pendekatan dan Jenis Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan mengintegrasikan analisis algoritma *clustering* dan model optimasi portofolio. Jenis penelitian yang dipakai adalah kuantitatif deskriptif yang memaparkan kinerja portofolio optimal menggunakan integrasi algoritma *clustering* terbaik dengan gabungan algoritma genetik dan *particle swarm optimization* dalam memberikan gambaran tentang strategi pemilihan aset dan pembobotan portofolio secara efektif. Oleh karena itu, peneliti menggunakan jenis data kuantitatif yang diperoleh dari perhitungan tingkat *return* harian saham periode 1 Januari 2022 hingga 31 Desember 2024 yang konsisten terdaftar dalam indeks LQ-45 selama periode Januari 2022 hingga Desember 2024. Selain itu, data kuantitatif yang digunakan adalah hasil perhitungan rasio ROE, *Debt-to-Equity*, dan *Cash Ratio* berdasarkan laporan keuangan tahunan periode 2024 dari perusahaan yang konsisten terdaftar dalam indeks LQ-45 selama periode Januari 2022 hingga Desember 2024. Terakhir, *risk-free rate* yang digunakan dalam perhitungan rasio Sharpe adalah *BI 7-Days Repo Rate* (BI7DRR) pada akhir periode penelitian, yakni Desember 2024 dan sebesar 6%. Untuk kemudahan dalam pengolahan dan analisis data, peneliti menggunakan bantuan perangkat lunak *Microsoft Excel* 2019 dan *R* versi 4.4.2.

2.2 Alur Kerja

Langkah-langkah penelitian secara terurut dibagi menjadi tiga tahap utama sebagai berikut:

1. *Pre-processing*
   Tahapan ini berisikan pengolahan data *daily closing price* dan laporan keuangan dari setiap saham yang terpilih. Alur *pre-processing* secara terurut sebagai berikut:
   1) Perhitungan *return* harian dari setiap saham
   2) Perhitungan mean dan standar deviasi dari *return* harian setiap saham
   3) Perhitungan rasio keuangan (*Return-On-Equity*, *Debt-to-Equity*, dan *Cash Ratio*) dari setiap saham
   4) *Exploratory data analysis* pada variabel mean dan standar deviasi dari *return* harian, serta rasio keuangan
   5) Transformasi data jika ditemukan data pencilan atau *outliers*
   6) Pengurangan jumlah dimensi data menggunakan *principal component analysis*

2. Klasterisasi

   Tahapan ini berisikan penentuan jumlah klaster optimal dan pengelompokan saham dari masing-masing algoritma *clustering* yang digunakan, yakni *K-Means*, DBSCAN, dan *Agglomerative* metode *Average Linkage*. *Silhouette Score* digunakan untuk menentukan jumlah klaster optimal dari setiap algoritma dan untuk menentukan algoritma klaster yang paling cocok untuk data. Alur *clustering* secara terurut sebagai berikut:

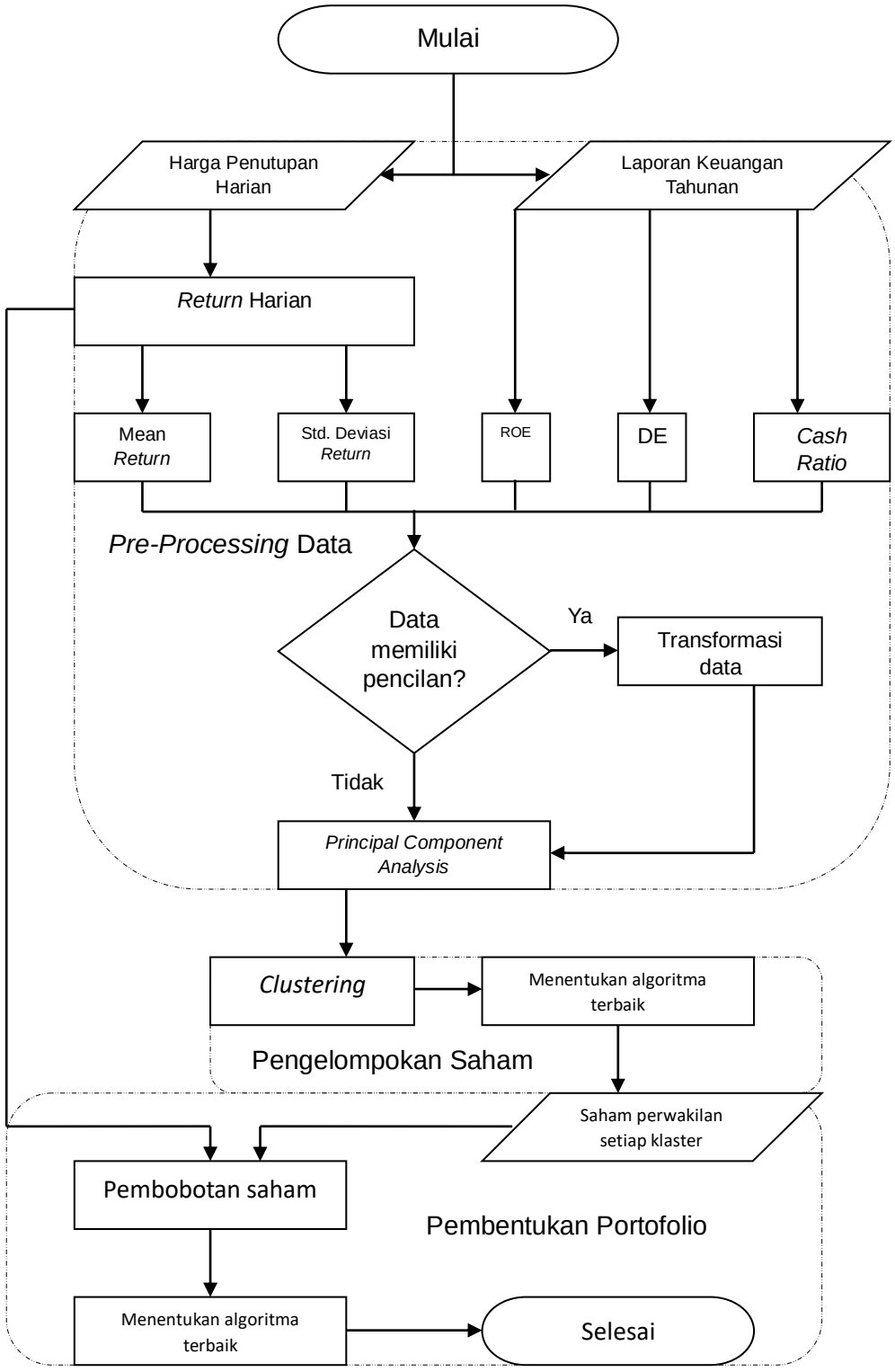
   1) Penentuan jumlah klaster optimal menggunakan algoritma *K-Means*
   2) Penentuan jumlah klaster optimal menggunakan algoritma *agglomerative* metode *average linkage*
   3) Penentuan jumlah klaster optimal menggunakan algoritma DBSCAN
   4) Penentuan algoritma *clustering* terbaik berdasarkan skor *silhouette* tertinggi

3. Pembentukan Portofolio

   Tahapan ini berisikan pemilihan saham sebagai perwakilan dari setiap klaster berdasarkan algoritma *clustering* terbaik. *Return* harian dari saham-saham yang terpilih akan digunakan untuk penentuan bobot saham sehingga diperoleh portofolio yang optimal. Kinerja portofolio yang dibentuk akan diukur menggunakan rasio Sharpe. Alur pembentukan portofolio saham sebagai berikut:

   1) Penentuan saham perwakilan di setiap klaster
   2) Pembobotan saham menggunakan algoritma *particle swarm optimization*
   3) Pembobotan saham menggunakan gabungan algoritma genetik dan *particle swarm optimization*
   4) Perbandingan kinerja portofolio menggunakan rasio Sharpe

Seluruh tahapan yang telah dipaparkan dapat digambarkan secara ringkas dalam *flowchart* pada Gambar 2.1.



Gambar 2.1 Alur kerja penelitian