

## BAB III

### METODE PENELITIAN

#### 3.1 Deskripsi Masalah

Penelitian ini membahas optimisasi portofolio saham dengan model yang digunakan adalah model MAD. Portofolio yang optimal merupakan portofolio dengan proporsi aset masing-masing saham dalam portofolio tersebut menghasilkan nilai *expected return* ( $E(R_i)$ ) tertentu dengan nilai risiko ( $MAD_i$ ) serendah-rendahnya. Untuk menyelesaikan masalah tersebut pada penelitian ini dilakukan optimisasi portofolio saham dengan menggunakan algoritma *Grey Wolf Optimizer* (GWO). Penelitian ini akan menggunakan data historis saham dari indeks saham IDX30.

#### 3.2 Studi Pustaka

Dalam penelitian ini, dilakukan studi pustaka dengan mempelajari teori portofolio saham termasuk model *Mean Absolute Deviation*, serta algoritma *Grey Wolf Optimizer* dari berbagai jurnal.

#### 3.3 Data Penelitian

Penelitian ini menggunakan data yang diperoleh dari laman <https://finance.yahoo.com/>. Data yang diambil dari laman tersebut adalah data harga penutupan saham mingguan dari berbagai saham yang termasuk ke dalam indeks saham IDX30 pada periode Januari – Desember 2023. Setelah itu, dilakukan pengolahan data sehingga dapat diperoleh nilai *return* saham, *expected return saham*, serta nilai risiko dari saham-saham tersebut.

#### 3.4 Model Optimisasi

Penelitian ini akan menggunakan model MAD. Persamaan (2.3) dan (2.6) akan digunakan sebagai fungsi tujuan dengan tujuan memaksimalkan nilai *expected return* terhadap nilai risiko portofolio dengan nilai minimum *expected return* dan nilai maksimum risiko sesuai preferensi investor. Kendala lain dari fungsi tujuan yang digunakan adalah bobot portofolio bernilai 1 dan bobot tiap saham sesuai dengan batas bawah dan batas atas bobot saham. Fungsi tujuan dari model MAD dapat dinyatakan sebagai :

Memaksimumkan

$$f(w) = \frac{\sum_{i=1}^n E(R_i)w_i}{\sum_{i=1}^n MAD_i w_i} \quad (3.1)$$

dengan kendala

$$\sum_{i=1}^n w_i = 1; \varepsilon_i \leq w_i \leq \vartheta_i \quad (3.2)$$

$$\sum_{i=1}^n E(R_i)w_i \geq R_{investor} \quad (3.3)$$

$$\sum_{i=1}^n MAD_i w_i \leq \sigma_{investor} \quad (3.4)$$

Adapun asumsi dalam penelitian ini adalah nilai risiko portofolio ( $\sigma$ ) tidak bernilai 0. Hal tersebut dilakukan agar tidak terjadi error dalam perhitungan nilai fungsi tujuan ( $f(w)$ ).

### 3.5 Penyelesaian Model Optimisasi Portofolio dengan Algoritma GWO

Terdapat beberapa tahapan dalam melakukan optimisasi portofolio dengan menggunakan algoritma GWO.

1. Mencari nilai *expected return* ( $E(R_i)$ ) dan nilai risiko ( $MAD_i$ ) saham;

Setelah didapatkan data nilai *return* ( $R_i$ ) dari saham-saham yang digunakan, akan dihitung nilai *expected return* tiap saham ( $E(R_i)$ ) menggunakan persamaan (2.2). Saham yang dimasukkan ke dalam portofolio adalah saham yang memiliki nilai *expected return* ( $E(R_i)$ ) positif. Hal tersebut dikarenakan saham yang memiliki nilai *expected return* ( $E(R_i)$ ) negatif dapat mengakibatkan nilai fungsi tujuan menjadi negatif dan tidak maksimal, sehingga portofolio tidak optimal.

2. Menginisialisasi parameter;

Tahap pertama yang dilakukan dalam algoritma GWO merupakan inisialisasi parameter. Algoritma GWO bekerja untuk melakukan optimisasi portofolio bergantung pada beberapa nilai parameter yang harus ditetapkan di awal proses. Parameter-parameter tersebut adalah sebagai berikut:

- 1) Maksimum iterasi;

Parameter ini menentukan berapa kali sekelompok serigala akan berpindah posisi dalam berburu.

2) Populasi serigala;

Parameter ini menentukan banyaknya serigala dalam satu kelompok berburu. Dalam optimisasi portofolio, serigala menggambarkan sebuah portofolio dan posisi serigala menggambarkan bobot tiap saham yang termasuk ke dalam portofolio tersebut.

3) Batas bawah dan batas atas bobot saham; dan

Parameter ini menentukan nilai minimum dan nilai maksimum bobot dari masing-masing saham.

4) Nilai  $R_{investor}$  dan  $\sigma_{investor}$ ;

Parameter ini menentukan nilai minimum *expected return* dan nilai maksimum risiko portofolio yang dihasilkan sesuai dengan preferensi masing-masing investor.

### 3. Pembangkitan Populasi Serigala

Selanjutnya dilakukan pembangkitan populasi acak untuk tiap SN serigala dengan menggunakan

$$w_{i,j} = w_j^{\min} + rand(0, 1) \times (w_j^{\max} - w_j^{\min}) \quad (3.5)$$

di mana  $w_{i,j}$  adalah bobot aset saham ke- $j$  pada portofolio ke- $i$ ,  $rand(0, 1)$  adalah angka acak dari 0 sampai 1,  $w_j^{\max}$  dan  $w_j^{\min}$  masing masing merupakan batas atas dan batas bawah dari saham ke- $j$ . Kemudian agar portofolio memenuhi kendala  $\sum_{i=1}^n w_i = 1$ , dilakukan perhitungan  $w_{i,j}^{new} = \frac{w_{i,j}^{old}}{\sum w_{i,j}^{old}}$  untuk tiap aset pada portofolio tersebut sehingga kendala terpenuhi.

### 4. Mengevaluasi seluruh serigala menggunakan fungsi tujuan

Setiap serigala akan dihitung nilai *expected return* ( $E(R_i)$ ) dan risiko ( $MAD_i$ ). Setelah itu, dilakukan perhitungan nilai fungsi tujuan dari setiap serigala. Agar portofolio yang terpilih adalah portofolio yang memenuhi *expected return* minimum dan risiko maksimum, maka perlu diberikan penalti apabila portofolio tersebut tidak memenuhi kendala tersebut. Setiap kendala yang dilanggar

bernilai 1 dan bernilai 0 jika tidak dilanggar. Perhitungan nilai *fitness* dapat menggunakan persamaan sebagai berikut.

$$fitness = \frac{f(w)}{1 + F_p} \quad (3.6)$$

dengan  $F_p$  merupakan nilai penalti.

5. Menentukan serigala alpha, beta, dan delta

Setelah dihitung nilai *fitness* dari setiap serigala, dipilih tiga serigala yang memiliki nilai *fitness* terbaik, yaitu:

$X_\alpha$  : agen pencarian terbaik (alpha).

$X_\beta$  : agen pencarian terbaik kedua (beta).

$X_\delta$  : agen pencarian terbaik ketiga (delta).

6. Memperbarui posisi serigala

Setiap serigala melakukan perpindahan posisi dengan memperhatikan posisi serigala alpha, beta, dan delta dengan menggunakan persamaan (2.12), (2.13), dan (2.14). Jika nilai *fitness* posisi serigala yang baru lebih baik dibandingkan posisi serigala yang lama, maka posisi serigala tersebut akan berpindah ke posisi serigala yang baru. Namun, jika nilai yang diperoleh lebih kecil atau sama, maka serigala tetap di posisi yang lama.

7. Menghitung nilai *fitness*

Setelah melakukan perpindahan posisi, setiap serigala akan dihitung kembali nilai *expected return* dan risiko. Setelah itu, dilakukan perhitungan nilai *fitness* dari setiap serigala.

8. Perbarui serigala alpha, beta, dan delta

Setelah dihitung nilai *fitness* dari setiap serigala, dipilih serigala alpha, beta, dan delta yang baru dengan ketentuan seperti di poin 6.

9. Perbarui nilai  $a$ ,  $A$ , dan  $C$  untuk iterasi selanjutnya

Untuk iterasi selanjutnya, perlu dilakukannya pembaruan nilai  $a$ ,  $A$ , dan  $C$ . Di mana nilai  $a$  berkurang secara linear dari 2 ke 0 selama berjalannya iterasi dan akan mempengaruhi nilai  $A$ . Sedangkan nilai  $C$  diacak setiap iterasinya pada rentang  $[0,2]$ .

10. Memilih serigala alpha sebagai solusi terbaik

Apabila maksimum iterasi telah terpenuhi, dipilih serigala alpha sebagai solusi optimum. Serigala alpha merupakan serigala dengan nilai *fitness* terbaik.

### 3.6 Contoh Implementasi

Algoritma *Grey Wolf Optimizer* akan digunakan dalam menyelesaikan model optimisasi yang dibentuk sebelumnya. Berikut adalah detail implementasi dari algoritma *Grey Wolf Optimizer* untuk optimisasi portofolio dengan contoh kasusnya. Data harga penutupan saham yang digunakan sebagai sampel terdapat pada Tabel 3.1.

Tabel 3.1 Data Harga Penutupan Saham

ADRO	BBNI	UNVR	TLKM
2680	9175	3660	3720
2890	9375	3610	3690
2840	9475	3500	3720
2950	9750	3850	3860
2850	10325	3740	3750

Sebelum masuk ke algoritma *Grey Wolf Optimizer*, akan dicari terlebih dahulu nilai *return* ( $R_i$ ) menggunakan persamaan (2.1). Dari data yang dikumpulkan, diperoleh nilai *return* ( $R_i$ ) tiap periode dari masing-masing saham yang disajikan pada Tabel 3.2.

Tabel 3.2 Nilai Return Sampel Saham

RETURN			
ADRO	BBNI	UNVR	TLKM
0,07836	0,0218	-0,0137	-0,0081
-0,0173	0,01067	-0,0305	0,00813
0,03873	0,02902	0,1	0,03763
-0,0339	0,05897	-0,0286	-0,0285

Setelah didapatkan nilai *return* ( $R_i$ ) dari masing-masing saham, dihitung nilai *expected return* ( $E(R_i)$ ) dengan menggunakan Persamaan (2.2). Jika ada saham yang memiliki nilai *expected return* negatif, maka saham tersebut tidak dimasukkan

ke dalam portofolio. Didapatkan nilai *expected return* dari empat sampel saham yang ditampilkan pada Tabel 3.3.

Tabel 3.3 *Return* Rata-rata Empat Sampel Saham

RETURN RATA-RATA	
ADRO	0,016472815
BBNI	0,030115784
UNVR	0,006824114
TLKM	0,002300641

Setelah didapatkan nilai *expected return* ( $E(R_i)$ ) dan dipilih saham yang dimasukkan ke dalam portofolio, dihitung nilai risiko ( $MAD_i$ ) dengan menggunakan Persamaan (2.5). Dapat dilihat nilai risiko dari empat sampel saham yang disajikan pada Tabel 3.4.

Tabel 3.4 Nilai Risiko Empat Sampel Saham

Risiko	
ADRO	0,04207249
BBNI	0,01442929
UNVR	0,04658794
TLKM	0,0205816

Selanjutnya adalah memulai tahapan algoritma *Grey Wolf Optimizer* sebagai berikut.

#### 1. Tahap Inisialisasi

Dilakukan tahapan inisialisasi dengan menentukan nilai maksimum iterasi, populasi serigala, batas bawah bobot saham, batas atas bobot saham, nilai  $R_{investor}$ , dan nilai  $\sigma_{investor}$ . Sebagai contoh, dilakukan inisialisasi parameter sebagai berikut:

IN	=	1	(Maksimum iterasi)
SN	=	5	(Populasi serigala)
$\varepsilon_i$	=	0,01	(Batas minimum bobot saham)
$\vartheta_i$	=	1	(Batas maksimum bobot saham)

$$R_{investor} = 0,01 \quad (\text{nilai minimum } \textit{expected return})$$

$$\sigma_{investor} = 0,03 \quad (\text{nilai maksimum risiko})$$

## 2. Pembangkitan Populasi Serigala

Selanjutnya dilakukan pembangkitan populasi acak untuk tiap serigala dengan jumlah serigala sebanyak SN. Setelah itu, hitung nilai *fitness* dari masing-masing portofolio dengan menggunakan persamaan (3.6). Kemudian didapatkan hasil inisialisasi awal dan nilai fungsi tujuan yang disajikan pada Tabel 3.5.

Tabel 3.5 Inisialisasi Populasi Serigala dan Nilai Fungsi Tujuan

	ADRO	BBNI	UNVR	TLKM	$f(w)$
Serigala 1	0,03167606	0,3530958	0,281180435	0,334048	0,524302
Serigala 2	0,39646455	0,1665103	0,425701322	0,011324	0,369786
Serigala 3	0,63881735	0,178355	0,030470235	0,152357	0,48383
Serigala 4	0,2879835	0,351092	0,0503227	0,310602	0,63178
Serigala 5	0,04995657	0,3890246	0,418017473	0,143001	0,521699

Setelah itu ditentukan kedudukan serigala alpha, beta, dan delta di mana serigala alpha adalah serigala yang memiliki nilai *fitness* tertinggi, serigala beta adalah serigala yang memiliki nilai *fitness* kedua tertinggi, serigala delta adalah serigala yang memiliki nilai *fitness* ketiga tertinggi, dan serigala yang lain merupakan serigala omega. Maka dapat dipilih Serigala 4 sebagai serigala alpha, Serigala 1 sebagai serigala beta, dan Serigala 5 sebagai serigala delta.

## 3. Pembaruan Posisi Serigala

Setelah ditentukan serigala alpha, beta, dan delta, dilakukan pembaruan posisi serigala. Sebagai contoh, Serigala 1 akan bergerak dengan memperhatikan posisi Serigala 4, Serigala 1, dan Serigala 5. Jika nilai *fitness* posisi Serigala 1 yang baru lebih besar dibandingkan posisi Serigala 1 yang lama, maka posisi Serigala 1 akan digantikan dengan nilai yang baru. Namun, jika nilai yang diperoleh lebih kecil atau sama, maka nilai Serigala 1 akan tetap. Hal ini berlaku untuk setiap serigala dan akan berlanjut pada iterasi selanjutnya. Didapatkan hasil perhitungan pembaruan posisi serigala yang disajikan pada Tabel 3.6.

Tabel 3.6 Hasil Perhitungan Pembaruan Posisi Serigala

	ADRO	BBNI	UNVR	TLKM	<i>Fitness</i>
Serigala 1	0,14833922	0,2897629	0,27336048	0,288537	0,470831
Serigala 2	0,23613987	0,3563619	0,157082287	0,250416	0,590584
Serigala 3	0,3696786	0,2396803	0,217119554	0,173522	0,46451
Serigala 4	0,13852844	0,448305	0,088263258	0,324903	0,741809
Serigala 5	0,12320538	0,3644041	0,249840202	0,26255	0,557136

Selanjutnya lakukan pembaruan posisi untuk serigala yang memiliki nilai *fitness* baru yang lebih besar dari nilai *fitness* yang sebelumnya. Pada iterasi ini, dilakukan pembaruan posisi untuk Serigala 2, Serigala 4, dan Serigala 5. Sedangkan, Serigala 1 dan Serigala 3 tidak memperbarui posisinya karena memiliki nilai *fitness* baru yang lebih kecil dari nilai *fitness* yang sebelumnya. Sehingga didapatkan nilai dari masing-masing serigala untuk iterasi selanjutnya yang ditampilkan pada Tabel 3.7.

Tabel 3.7 Posisi Serigala Baru

	ADRO	BBNI	UNVR	TLKM	<i>Fitness</i>
Serigala 1	0,03167606	0,3530958	0,281180435	0,334048	0,524302
Serigala 2	0,23613987	0,3563619	0,157082287	0,250416	0,590584
Serigala 3	0,63881735	0,178355	0,030470235	0,152357	0,48383
Serigala 4	0,13852844	0,448305	0,088263258	0,324903	0,741809
Serigala 5	0,12320538	0,3644041	0,249840202	0,26255	0,557136

Kemudian, *update* serigala  $\alpha$ ,  $\beta$ , dan  $\delta$ . Pada kasus ini, serigala alpha adalah Serigala 4, serigala beta adalah Serigala 2, dan serigala delta adalah Serigala 5. Setelah itu, *update* nilai  $a$ ,  $A$ , dan  $C$  untuk iterasi selanjutnya, kemudian lakukan pembaruan posisi hingga maksimum iterasi terpenuhi.

#### 4. Pemilihan Solusi Terbaik

Setelah maksimum iterasi terpenuhi, dilakukan pemilihan solusi terbaik dengan memilih serigala dengan nilai *fitness* tertinggi. Dalam kasus ini, nilai *fitness*



tertinggi terdapat pada Serigala 4 dengan nilai  $f(w) = 0,741809$  dengan bobot sebagai berikut:

Tabel 3.8 Bobot Portofolio Optimal

ADRO	ANTM	BBNI	ASII
0,13853	0,44831	0,08826	0,3249

Dengan bobot yang diperoleh, didapatkan nilai *expected return* = 0,017132 dan nilai risiko = 0,023095.

### 3.6 Validasi Program Perhitungan

Diperlukan waktu yang lama untuk melakukan perhitungan dan implementasi algoritma GWO secara manual. Oleh karena itu, untuk mempercepat perhitungan dan implementasi, pada penelitian ini dibuat program dengan menggunakan *Python*. Dipilih bahasa pemrograman *Python* karena ketersediaan pustaka (*library*) yang banyak dan kemudahan implementasinya.

Untuk mengukur kualitas dan efektivitas program yang dibuat, dilakukan validasi dengan menggunakan fungsi *benchmark*. Fungsi *benchmark* yang akan digunakan untuk melakukan validasi adalah fungsi Sphere dan Rastrigin. Jika diperoleh solusi yang mendekati nilai optimum global maka tahapan akan dilanjutkan ke implementasi. Jika solusi yang diperoleh jauh dari nilai optimum global maka tahapan akan diulang dari pembentukan program.

Dibuat *Graphical User Interface* (GUI) untuk program yang digunakan pada penelitian ini. Dengan adanya GUI, investor dapat lebih mudah dalam memasukan data yang dimiliki oleh investor tersebut dan nilai parameter yang diinginkan. Dengan demikian, GUI dapat meningkatkan aksesibilitas investor dalam mencari portofolio yang optimal.

### 3.7 Penarikan Kesimpulan

Tahap terakhir dari penelitian ini yaitu dilakukan penarikan kesimpulan untuk menjawab pertanyaan pada rumusan masalah dari penelitian ini.