

Optimasi Portofolio Mean-semivariance dengan Algoritma Genetika Multiobjective Evolutionary NSGA II

Anjar Pratiwi¹, Deni Saepudin², Rian Febrian Umbara³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹anjarpratiwi@students.telkomuniversity.ac.id, ²denisaepudin@telkomuniversity.ac.id,

³rianum@telkomuniversity.ac.id,

Abstrak

Dalam investasi saham seorang investor sebaiknya membuat portofolio optimal agar memperoleh hasil yang memuaskan dengan nilai *return* tinggi atau nilai risiko rendah. Maka dari itu sebagai investor harus dapat berkompromi dalam menangani 2 objektif, objektif yang dimaksud adalah objektif 1 *expected return* sedangkan untuk objektif 2 adalah risiko dan untuk menyelesaikan *problem* tersebut dapat menggunakan algoritma *Multiobjective NSGA (Non - Dominated Shorting Genetic Algorithm)*. Sudah ada beberapa penelitian yang terkait dengan Algoritma *Multiobjective NSGA – II* dan terbukti bahwa algoritma ini merupakan algoritma yang cukup baik untuk menangani *problem* optimasi 2 objektif. Data yang digunakan pada tugas akhir ini adalah data saham yang tergabung dalam *index LQ45*. Hasil akhir dari penerapan Algoritma Genetika *Multiobjective NSGA-II* akan menghasilkan bobot yang nantinya bobot tersebut akan digunakan untuk menghitung *return* portofolio dan risiko kemudian akan membentuk *efficient frontier*. Pada penelitian ini risiko menggunakan *semivariance* terbukti menghasilkan risiko yang optimal jika dibandingkan dengan *variance*.

Kata kunci : NSGA-II , *Multiobjective* , *Optimasi portofolio* , *semivariance*, *efficient frontier*

Abstract

In stock investment an investor should make optimal portfolio in order to obtain satisfactory results with high return value or low risk value. Therefore, as an investor must be able to compromise in dealing with 2 objectives, the intended purpose is objective 1 which is expected return while for objective 2 is the Risk and to solve the that problem can be used Multiobjective NSGA (Non-Dominated Shorting Genetic Algorithm) algorithm. Therefore, as an investor must be able to compromise in dealing with 2 objectives, the intended purpose is objective 1 which is expected again while for objective 2 is the problem and to solve the problem that can be used Multiobjective NSGA (Non-Dominated Shorting Genetic Algorithm) algorithm. There have been several studies related to the Algoritma Multiobjective NSGA-II and it is evident that this algorithm is a pretty good recipient for the problem of 2 goal optimization problems. The data used at this time is the data belonging to the LQ45 index. The final result of the application of the Multi-objective Genetic Algorithm of NSGA-II will result in a weight that the weight will be used to calculate portfolio return and risk using semivariance then establish an efficient frontier. In this study the risk of using semivariance proved to produce an optimal risk when compared with the variance.

Keywords: NSGA-II , *Multiobjective* , *Optimasi portofolio* , *semivariance*, *efficient frontier*

1. Pendahuluan

1.1 Latar Belakang

Investasi adalah menanamkan modal ke berbagai sektor dengan tujuan mendapatkan jumlah keuntungan yang diharapkan pada masa yang akan datang. Tentunya dalam berinvestasi pasti akan terjadi timbal balik, jika orang tersebut ingin mendapatkan *return* yang besar maka risiko yang ditanggung juga besar, begitupula sebaliknya jika orang tersebut takut akan mengambil risiko yang besar dan lebih memilih risiko yang kecil maka akan mendapatkan *return* yang kecil juga, hal tersebut menimbulkan *trade-off* yang merupakan hubungan antara *return* dan risiko. Konsep *trade-off* tersebut dicetuskan oleh ahli dasar ilmu keuangan yang termuka Markowitz, dengan adanya pertimbangan *trade-off* tersebut optimasi portofolio setiap orang mungkin akan berbeda – beda. Maka seorang yang akan berinvestasi harus bersikap realistis dalam menginvestasikan dananya dan memilih portofolio yang sesuai dengan tujuan investor dalam berinvestasi.

Optimasi adalah proses memaksimalkan atau meminimasi suatu fungsi tujuan. Dalam Tugas ini masalah yang akan dibahas adalah bagaimana menentukan himpunan porotofolio menggunakan model *mean-semivariance* untuk menyelesaikan optimasi portofolio saham dengan 2 fungsi tujuan. Optimasi portofolio diperlukan

untuk mendapatkan *return* dan risiko yang optimal. Oleh karena itu ada himpunan portofolio yang merupakan kumpulan dari kedua objektif, objektif 1 adalah risiko dan objektif 2 adalah *return*. Berdasarkan pertimbangan tersebut selanjutnya akan dicari himpunan portofolio yang merupakan kompromi dari kedua objektif ini.

Pada tugas akhir ini untuk menyelesaikan permasalahan optimasi 2 objektif sekaligus yaitu menggunakan algoritma *Multiobjective Evolutionary NSGA II (Non- Sorting Dominated Genetic Algorithm II)* yang merupakan bagian dari algoritma genetika. Algoritma *multiobjective NSGA-II (Non-dominated sorting Genetic Algorithm II)* merupakan salah satu algoritma yang populer digunakan untuk permasalahan optimisasi. Algoritma NSGA II pertama kali disajikan dalam Deb et al., (2002) dan muncul sebagai perbaikan pada NSGA generasi pertama. Sudah ada beberapa penelitian yang dilakukan untuk menyelesaikan masalah optimasi *multiobjective*, diantaranya adalah Kalyanmoy Deb yang membahas tentang *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*. Hasil dari penelitian yang sudah dilakukan tersebut menunjukkan bahwa algoritma *Multiobjective NSGA – II* dapat menyelesaikan masalah optimasi *multiobjective*. Proses kerja algoritma genetika NSGA – II mirip dengan proses algoritma genetika asli perbedaan yang mencolok terdapat pada proses seleksinya. Algoritma genetika NSGA – II akan menyimpan solusi yang optimal dari tiap generasi, kemudian solusi yang optimal tersebut akan diseleksi solusi mana yang *non-dominance* berdasarkan *front*, maksud dari *non-dominance* adalah solusi yang tidak di dominasi oleh solusi lain. Solusi dari hasil seleksi tersebut selanjutnya akan dibandingkan dengan solusi – solusi dari setiap generasi. Pada tugas akhir ini algoritma genetika *multiobjective NSGA – II* digunakan untuk mendapatkan individu terbaik yang berupa bobot masing – masing saham pada portofolio, bobot yang telah didapatkan dari algoritma tersebut kemudian akan digunakan untuk menghitung *return* dan risiko portofolio saham dan menghasilkan grafik *efficient frontier*.

1.2 Topik dan Batasannya

Dalam tugas akhir ini topik yang dianalisis yaitu mendapatkan portofolio saham yang optimal bagi investor. Batasan masalah pada tugas akhir ini dalam portofolio terdapat 5 saham yang dioptimasi. Ada banyak metode yang digunakan untuk mendapatkan optimasi portofolio saham, karena pada penelitian ini menggunakan 2 objektif maka dari itu penulis memilih untuk menggunakan algoritma *Multiobjective Non-dominated Sorting Genetic Algorithm (NSGA II)*. Data history saham yang digunakan adalah data saham yang tergabung dalam indeks LQ45, data tersebut diperoleh dari *finance.yahoo.com* dan data yang digunakan adalah harga penutupan saham data mingguan / *weekly* dalam kurun waktu selama 5 tahun (2012-2017).

1.3 Perumusan Masalah

Perumusan masalah pada tugas akhir ini sebagai berikut:

1. Bagaimana cara menerapkan algoritma *Multiobjective Evolutionary NSGA II (Non- Sorting Dominated Genetic Algorithm II)* pada optimasi portofolio saham?
2. Bagaimana mencari portofolio saham yang optimal dengan model *mean-semivariance* yang merupakan kompromi dari 2 objektif yaitu *expected return* dan risiko?

1.4 Tujuan

Tujuan pada tugas akhir ini yaitu :

1. Mendapatkan bobot portofolio menggunakan Algoritma *Multiobjective (NSGA – II)*.
2. Mendapatkan sekumpulan solusi yang optimum dari 2 objektif *expected return* dan risiko yang membentuk *efficient frontier*.

2 Studi Terkait

2.1 Saham (Stock)

Saham (*stock*) didefinisikan sebagai surat berharga yang menunjukkan bagian kepemilikan seseorang pada suatu perusahaan atau perseroan terbatas. Ada 2 keuntungan yang didapatkan ketika seseorang investor membeli saham yaitu *Dividen* dan *Capital Gain*. *Dividen* adalah pembagian laba kepada pemegang saham (investor) berdasarkan banyaknya saham yang dimiliki, pembagian *dividen* diberikan pada akhir tahun periode pembukuan perusahaan. Sedangkan *Capital Gain* adalah selisih dimana harga jual lebih tinggi dari harga beli.

2.2 Return, Expected Return dan Risiko

2.2.1 Return

Return merupakan selisih yang didapatkan atas investasi yang dilakukan oleh investor. Cara untuk menghitung *return* harga saham yaitu dengan rumus [3]:

$$R_i = \frac{S_i - S_{i-1}}{S_{i-1}} \quad (1)$$

Keterangan :

R_i : Nilai *return* pada waktu ke i ,
 S_i : Harga saham pada waktu ke i .

2.2.2. Expected Return Saham

Expected return adalah nilai keuntungan yang diharapkan oleh investor setelah melakukan investasi. *Expected return* didapatkan dengan menghitung rata – rata saham dan dituliskan kedalam formula sebagai berikut:

$$\mu = \frac{\sum_{i=1}^T R_i}{T} \quad (2)$$

Keterangan :

μ : Nilai *expected return* saham,
 T : Jumlah periode waktu pengamatan (minggu),
 R_i : Nilai *return* saham pada jangka waktu ke i .

2.2.3. Risiko

Risiko dapat dinyatakan sebagai ukuran timbal balik dari *return* yang di dapatkan. Semakin tinggi *return* maka semakin tinggi pula risikonya

2.2.3.1. Variansi (Variance)

Menghitung nilai risiko dapat menggunakan variansi (*variance*). *Variance* menunjukkan seberapa besar *variable random*. Semakin besar sebarannya maka semakin besar pula risikonya, untuk menghitung variansi yaitu [8]:

$$\hat{\sigma}^2 = \frac{1}{n-1} E[R - E(R)]^2 \quad (3)$$

Keterangan :

$\hat{\sigma}^2$: Nilai variansi,
 E : *Expected value*,
 R : Nilai *return*.

2.2.3.2. Semivariansi (Semivariance)

Dengan menggunakan ukuran *semivariance* saham kemudian menghitung *semicovariance* antar saham dan bobot masing-masing saham dapat untuk menghitung besarnya risiko portofolio. Matriks *semivariance-semicovariance* dibentuk dari *semivariance*. Secara matematis, rumus untuk menghitung risiko portofolio menggunakan *mean-semivariance* adalah sebagai berikut [1]:

$$\hat{S} = E(\min(0, R - E(R)))^2 \quad (4)$$

Keterangan :

\hat{S} : Nilai *semivariance*,
 E : *Expected value*,
 R : Nilai *return*,
 $E(R)$: Nilai yang dijadikan *benchmark* atau nilai tolak ukur.

Kemudian untuk menghitung *semicovariance* menggunakan cara sebagai berikut [2]:

$$S_{pc} = \frac{1}{T} \sum_{t=1}^T [\text{Min}(0, R_{ta} - E(R)), \text{Min}(0, R_{tb} - E(R))] \quad (5)$$

Keterangan :

Spc : Nilai *semicovariance*,
 T : Jumlah periode waktu,
 R_{ta} : Nilai *return* saham a,
 R_{tb} : Nilai *return* saham b,
 E(R) : Nilai Brechmark atau nilai tolak ukur.

2.3 Porftfolio Saham

Portofolio saham adalah kumpulan aset investasi yang berupa saham, baik yang dimiliki oleh perorangan maupun perusahaan dengan bobot atau proporsi saham yang berbeda - beda sesuai dengan apa yang diharapkan oleh investor itu sendiri. Bobot dilambangkan dengan (w₁,w₂,w₃,...wn) . Jika seluruh bobot dijumlahkan maka totalnya akan berjumlah 100% atau 1.

2.3.1 Return dan Expected Return Portofolio

Semua investor saat hendak melakukan investasi tentunya mengharapkan keuntungan yang besar, nilai kembalian pada portofolio disebut *return* portofolio sedangkan *expected return* portofolio adalah nilai harapan dari *return* portofolio. Secara matematis rumus untuk menghitung *return* dan *expected return* portofolio dengan bobot w didefinisikan sebagai berikut [3]:

$$R_{pt} = \sum_{i=1}^n R_{it} W_i \quad (6)$$

$$E(R_p) = \frac{\sum_{i=1}^T W_i \mu_i}{T} \quad (7)$$

Keterangan :

R_{pt} : Nilai *return* portofolio pada waktu ke t,
 R_{it} : Nilai *return* saham ke i pada waktu ke t,
 R_p : Nilai *expected return* portofolio,
 W_i : Nilai bobot saham ke i,
 μ_i : Nilai *expected return* saham ke i,
 T : Jumlah periode waktu.

2.3.2 Risiko Portofolio

Risiko portofolio merupakan kebalikan dari *expected return* portofolio. Jika investor menginginkan nilai kembalian yang besar, maka risiko adalah nilai yang menyimpang dari apa yang diharapkan oleh investor. Berikut adalah persamaan untuk mendapatkan risiko portofolio :

$$S_p = \frac{1}{T} \sum_{t=1}^T (\min(0, R_p - E(R_p)))^2 \quad (8)$$

Keterangan :

S_p : Nilai risiko portofolio menggunakan *mean-semivariance*,
 T : Jumlah periode waktu,
 R_p : Nilai *return* portofolio,
 E(R_p) : Nilai Brechmark atau nilai tolak ukur.

2.3.3 Metode Portofolio Mean – Semivarince

Pada metode *semivariance*, untuk mendapatkan bobot saham yang menghasilkan resiko minimum dapat menggunakan persamaan [3] :

$$\mathbf{W} = \frac{\underline{\Delta} \mathbf{S}^{-1} \underline{\Delta} \mathbf{u}}{\underline{\Delta} \mathbf{u} \underline{\Delta} \mathbf{S}^{-1} \underline{\Delta} \mathbf{u}^T} \quad (9)$$

Keterangan:

- W** : Nilai bobot portofolio,
- $\underline{\Delta u}$: Nilai matrix kolom elemen (1xn),
- n : Jumlah banyaknya saham,
- $\underline{\Delta s}^{-1}$: Nilai matrix invers *semivariance-semicovariance*,
- $\underline{\Delta u}^T$: Nilai matrix kolom elemen (1xn) transpose.

Kemudian untuk mendapatkan bobot saham dengan nilai *semivariance* minimum adalah dengan memasukan *expected return* portofolio, dapat dirumuskan sebagai berikut [3] :

$$|W| = \frac{\begin{vmatrix} 1 & \underline{\Delta s}^{-1} \underline{\Delta u} & m^T \\ E(R_p) & m \underline{\Delta s}^{-1} & m^T \end{vmatrix} \begin{vmatrix} \underline{\Delta u} & \underline{\Delta s}^{-1} \\ m \underline{\Delta s}^{-1} & \underline{\Delta u}^T \end{vmatrix} + \begin{vmatrix} \underline{\Delta s}^{-1} & \underline{\Delta u}^T & 1 \\ m \underline{\Delta s}^{-1} & \underline{\Delta u}^T & E(R_p) \end{vmatrix}}{\begin{vmatrix} \underline{\Delta u} & \underline{\Delta s}^{-1} & \underline{\Delta u}^T \\ m \underline{\Delta s}^{-1} & \underline{\Delta u}^T & m \underline{\Delta s}^{-1} \underline{\Delta u}^T \end{vmatrix}} \quad (10)$$

Keterangan:

- [W] : Determinan matrix bobot portofolio
- $E(R_p)$: Nilai adalah *expected return* portofolio,
- m^T : Nilai matrix (nx1) *expected return transpose*,
- $\underline{\Delta s}^{-1}$: Nilai invers matrix *semivariance-semicovariance*,
- $\underline{\Delta u}$: Nilai matrix kolom elemen,
- $\underline{\Delta u}^T$: Nilai matrix kolom dengan elemen (1xn) *transpose*.

2.4 Optimasi Multiobjective

Optimasi adalah suatu proses untuk mencapai hasil yang ideal atau optimum. Pada *singleobjective* hanya ada satu fungsi tujuan yang akan dioptimalkan, berbeda dengan *multiobjective*. Optimasi *multiobjective* merupakan proses mengoptimalkan lebih dari satu fungsi tujuan sekaligus. Masalah *multiobjective* dapat didefinisikan sebagai berikut [4]

$$\min = [f_1(x), f_2(x), \dots, f_k(x)] \quad (11)$$

Masalah optimasi portofolio saham dengan pendekatan *mean-semivariance* dapat dituliskan sebagai masalah optimasi *multitobjective* yaitu [1] :

$$\min S = \sum_{i=1}^n W_i S_{pc} \quad (12)$$

$$\max E = \sum_{i=1}^n W_i E(\mu_i) \quad (13)$$

Dengan kendala :

$$\sum_{i=1}^n W_i = 1 \quad (14)$$

Keterangan:

- $\min S$: Nilai minimum risiko,
- $\max E$: Nilai maksimum return,
- n : Banyaknya saham dalam portofolio,
- S_{pc} : Nilai *semicovariance*,
- W_i : Nilai bobot saham i,
- $E(\mu_i)$: Nilai adalah *expected return* saham i,
- k : Jumlah fungsi tujuan,
- x : Vektor dari variabel keputusan $[x_1, x_2, \dots, x_n]^T$.

Ada 2 fungsi objektif yang akan dioptimalkan yaitu memaksimalkan return seperti yang tertera pada persamaan (8) dan meminimumkan Risiko seperti yang tertera pada persamaan (9). Untuk meminimumkan kedua fungsi tersebut dapat dituliskan rumus sebagai berikut :

$$\min f_1 = \sum_{i=1}^n W_i S_{pc} \quad (15)$$

$$\min f_2 = -(\sum_{i=1}^n W_i E(\mu_i)) \quad (16)$$

Keterangan:

- f_1 : Fungsi objektif 1
 f_2 : Fungsi objektif 2.

2.5 Multiobjective Non-dominated Sorting Genetic Algorithm II (NSGA – II)

Algoritma *multiobjective Non-dominated Sorting Genetic Algorithm* (NSGA- II) merupakan salah satu algoritma genetika yang dapat digunakan untuk menyelesaikan lebih dari satu fungsi objektif atau *multiobjective*. Algoritma NSGA – II ini adalah versi yang dimodifikasi dan dikembangkan, dimana lebih baik dalam *sorting algorithm* yang disertai dengan *elitism* [5]. Dalam permasalahan *multiobjective optimization* solusi terbaik ditentukan oleh *dominated solution set*. Himpunan (set) dari *non-dominated solutions* disebut *non-dominated set*. Himpunan ini memenuhi dua kondisi berikut jika [6]:

1. Setiap pasang solusi dalam himpunan ini tidak boleh saling mendominasi.
2. Setiap solusi yang berbeda diluar himpunan didominasi paling sedikit oleh satu anggota himpunan.

Perbedaan yang paling mencolok NSGA – II dari algoritma genetika biasa adalah adanya operator *crowding distance*. Cara kerja pada algoritma genetika NSGA-II semua individu dalam populasi gabungan yaitu orangtua (*parents*) dan anak (*child*) diurutkan berdasarkan solusi yang tidak mendominasi (*non-dominat*) pada setiap *front*, kemudian diklasifikasikan berdasarkan *non-dominance rank* ke dalam kelas, atau *front* yang berbeda. *Front* pertama yang terbentuk didasarkan pada kumpulan yang tidak mendominasi (*non- dominance*) dalam populasi awal, sementara solusi yang terburuk akan diletakan pada *front* paling akhir. Selanjutnya dilakukan penambahan operator *crowding distance* (jarak kerapatan antara titik populasi satu dengan titik populasi lainnya) dihitung oleh setiap individu. Tujuannya adalah untuk mempertahankan keragaman populasi, semakin besar nilai *crowding distance* maka akan menghasilkan populasi yang beragam. Orangtua (*parents*) selanjutnya diseleksi dari suatu populasi berdasarkan *rank* dan *crowding distance* menggunakan *binary tournament selection*. Pada proses ini, individu yang terpilih adalah individu dengan nilai *rank* yang lebih kecil dibandingkan dengan individu lain atau individu yang memiliki nilai *crowding distance* yang lebih besar dari individu yang lain. Populasi yang telah diseleksi selanjutnya melalui proses kawin silang (*crossover*) dan mutasi sehingga menghasilkan anak (*offspring*). Populasi awal yang berisikan orangtua dan anak kemudian diurutkan kembali berdasarkan individu yang tidak mendominasi individu lain (*non-dominance*).

2.5.1 Inisialisasi Populasi

Langkah awal pada algoritma NSGA- II yaitu inisialisasi populasi. Populasi adalah kumpulan dari beberapa individu atau kromosom, individu pada tugas akhir ini adalah bobot-bobot portofolio. Pada tahap ini yang dilakukan adalah membangkitkan populasi yang berisikan kromosom yang bernilai *real* secara acak (*random*), selanjutnya populasi yang telah dibangkitkan selanjutnya diurutkan berdasarkan yang tidak mendominasi (*non – dominated*).

2.5.2 Non-dominated Sort

Algoritma pengurutan *non-dominated solution set* sebagai berikut [6] :

- Untuk setiap individu p dalam populasi utama P, dilakukan :
 1. Inisialisasi $\mathcal{S}_p = \emptyset$, yang akan diisi oleh semua individu – individu yang didominasi oleh p.
 2. Inisialisasi $n_p = 0$, yang merupakan jumlah individu – individu yang mendominasi p.
 3. Kondisi untuk setiap individu q di dalam p,
 - a. Jika p mendominasi q pada sekumpulan \mathcal{S}_p yaitu : $\mathcal{S}_p = \mathcal{S}_p \cup \{q\}$
 - b. Jika q mendominasi p maka : $n_p = n_p + 1$
 4. Jika $n_p = 0$ artinya tidak ada individu yang mendominasi p, sehingga p merupakan *front* yang pertama kemudian individu p diberi rank 1. $p_{rank} = 1$. Perbarui *front* yang pertama dengan menambahkan pada *front* yang pertama, seperti ini $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$.
- Inisialisasi *front* = 1 $\rightarrow i = 1$.

Dilakukan pada *front* ke – i ketika $\mathcal{F}_i = \emptyset$

 1. $\mathcal{Q} = \emptyset$ yang merupakan penyimpanan kumpulan individu – individu untuk *front* ke (i+1).
 2. Untuk setiap individu p di dalam *front* \mathcal{F}_i yaitu :
 - setiap individu q yang berada dalam \mathcal{S}_p berisikan semua individu – individu yang didominasi oleh p.

- $nq = nq - 1$, pengurangan individu q.
- Jika $nq = 0$, maka tidak terdapat individu – individu pada *front* berikutnya yang didominasi oleh q. Sehingga lakukan ubah $q_{rank} = i + 1$ dan update Q dengan cara menambahkan individu q yaitu, $Q = Q \cup \{q\}$.
- 3. Tambahkan count *front* dengan cara $i = i + 1$.
- 4. Kemudian ubah Q menjadi *front* berikutnya, sehingga $\mathcal{F}_i = Q$.

2.5.3 Crowding Distance

Langkah selanjutnya adalah menghitung *crowding distance*. *Crowding distance* adalah jarak (*distance*) kerapatan antara *front* satu dengan *front* tetangga terdekatnya. Perhitungan nilai ini dilakukan hanya pada *front* yang tidak mendominasi (*non-dominated*) yang sama. Cara melakukan *crowding distance* sebagai berikut [7]:

1. Inisialisasi jarak untuk semua individu dengan nilai 0.
2. Untuk setiap fungsi tujuan m
 - a. Urutkan semua individu – individu di dalam *front* I berdasarkan tujuan m yaitu

$$I = \text{sort}(I, m)$$
 - b. Kemudian beri nilai jarak (*distance*) untuk setiap individu dalam *front* I sama dengan tak terhingga yaitu : $I[1]_{distance} = I[1]_{distance} = \infty$
 - c. Untuk $k=2$ sampai $(n-1)$
 - d. $I(d_k) = I(d_k) + \frac{1(k+1).m-1(k-1).m}{f_m^{max} - f_m^{min}}$ (17)
 - e. $I(k).m$ adalah nilai fungsi tujuan ke-m dari individu ke-k di I .

2.5.4 Binary Tournament Selection

Pada tahap *binary tournament selection* semua individu diseleksi, ada dua kriteria individu yang lolos pada tahap seleksi ini yaitu :

- a. *Non-domination rank*
Jika ada solusi dengan nilai *rank* berbeda maka solusi dengan nilai *rank* terkecil yang terpilih.
- b. *Crowding Distance*
Jika solusi yang didapatkan memiliki nilai *rank* yang sama maka solusi dengan nilai *crowding distance* terbesar yang terpilih.

2.5.5 Crossover dan Mutasi

Crossover (kawin silang) adalah proses dimana 2 individu (*parents*) yang berbeda dilakukan persilangan dan menghasilkan anak atau individu baru (*offspring*). *Crossover* untuk bilangan *real* dapat menggunakan *intermediate crossover*. Individu baru yang dihasilkan dapat dilihat pada persamaan berikut:

$$\begin{aligned} \text{Anak}_1 &= OT_1 + \alpha (OT_2 - OT_1) \\ \text{Anak}_2 &= OT_2 + \alpha (OT_1 - OT_2) \end{aligned} \quad (18)$$

Keterangan:

OT_1 : Orang Tua (*parents*) 1,

OT_2 : Orang Tua (*parents*) 2,

α : Parameter yang dibuat konstan atau di tentukan secara acak (*random*).

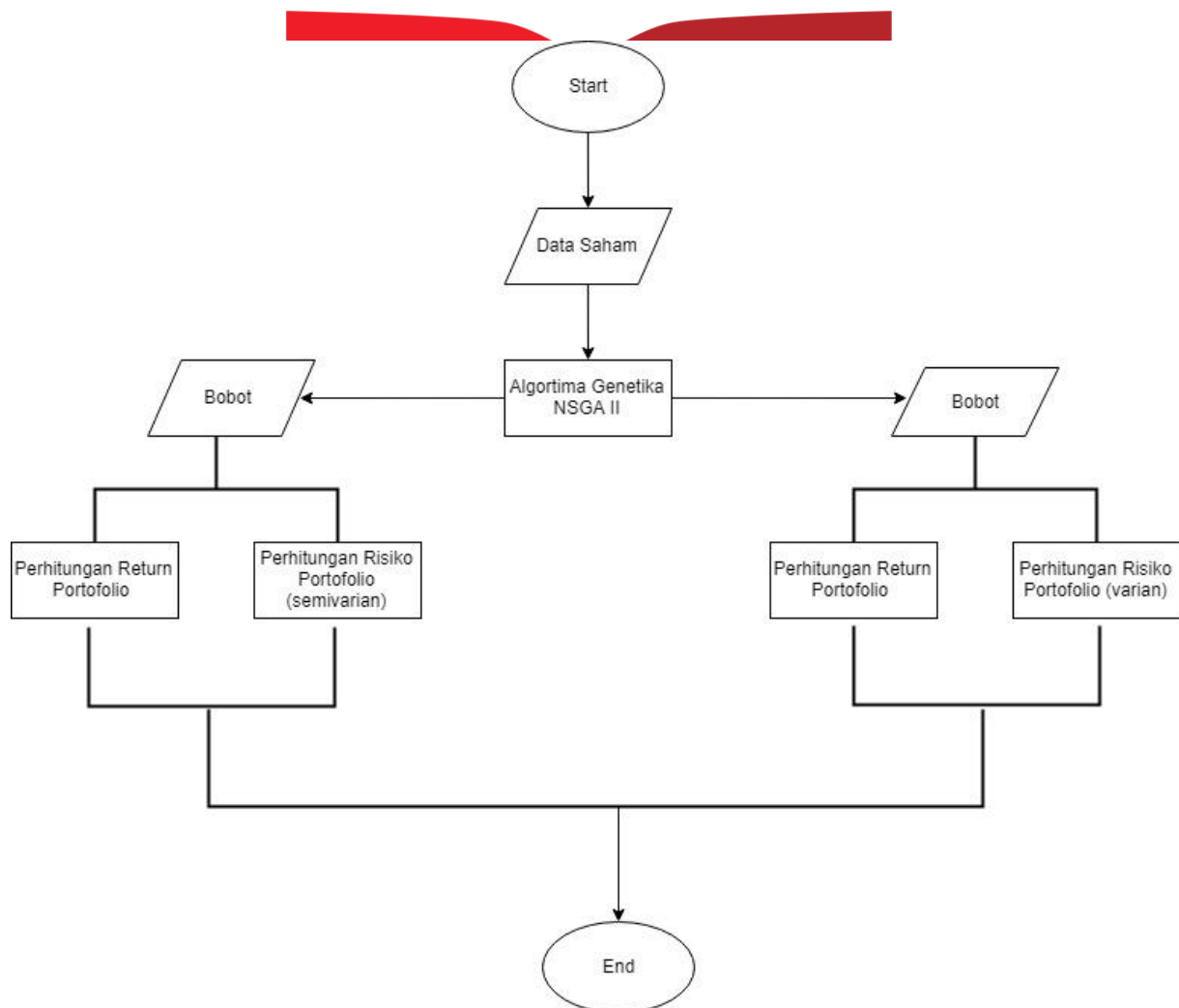
Setelah proses *crossover* selesai, tahap selanjutnya adalah mutasi. Mutasi yang dilakukan dalam tugas akhir ini menggunakan *swap mutation* cara kerjanya yaitu menukar gen yang dipilih secara acak dengan gen sesusudahnya.

2.6 Efficient frontier

Efficient frontier adalah garis imajiner dari sekumpulan portofolio atau kombinasi aset - aset yang memberikan *expected return* maksimum pada tingkat resiko tertentu. Portofolio yang tidak berada pada garis *efficient frontier* adalah portofolio yang tidak *efficient* karena pada tingkat resiko tertentu ada portofolio yang lain memberikan *return* yang lebih besar atau ada portofolio lain dengan hasil *return* yang sama namun memberikan resiko yang lebih kecil. *Efficient frontier* diperkenalkan pertamakali pada tahun 1995 oleh Harry Markowitz ekonom asal Amerika Serikat. Markowitz mengembangkan teori portofolio yang melihat pada bagaimana laba investasi dapat dioptimalkan.

3. Sistem yang Dibangun

3.1. Berikut adalah *flowchart* dari Optimasi Portofolio *Multiobjective* menggunakan NSGA – II

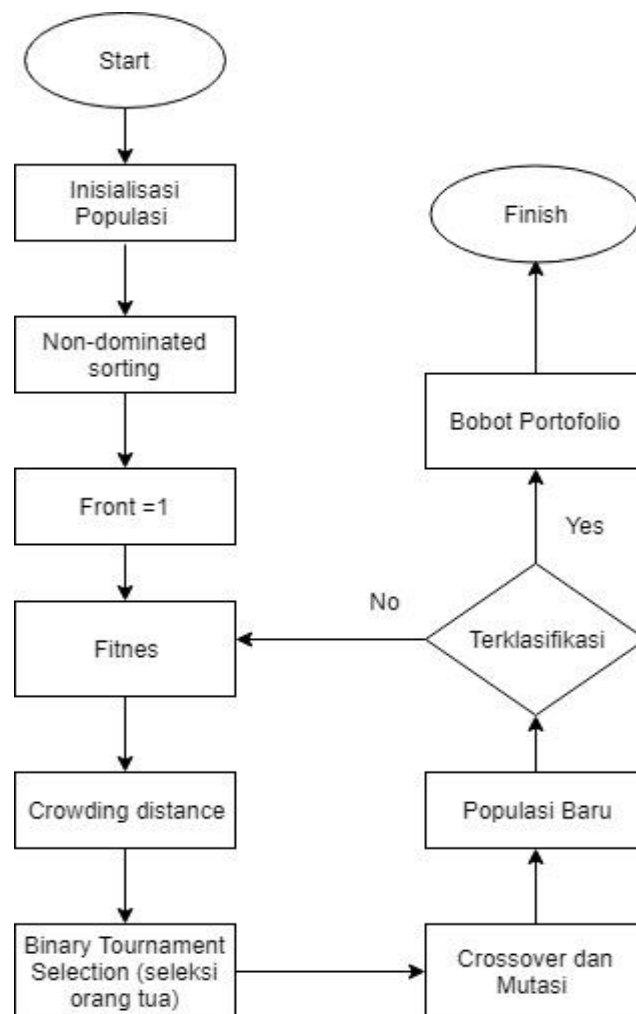


Gambar 3.1 Perancangan Sistem Optimasi Portofolio *Multiobjective*

- a. Input data harga saham
Data yang digunakan adalah harga penutupan/*close* saham data mingguan/*weekly* dari 5 saham yang didapatkan dari *index* LQ45 kurun waktu selama 5 tahun (2012-2017).
- b. Penerapan Algoritma Genetika *Multiobjective Evolutionary* NSGA-II untuk mendapatkan bobot portofolio.
menggunakan dengan objektif 1 dan objektif 2.
- c. Output bobot tiap saham
Algoritma NSGA-II akan menghasilkan bobot tiap – tiap saham.
- d. Perhitungan nilai *expected return* dan risiko portofolio.
Tahap selanjutnya hitung nilai *expected return* portofolio dengan menggunakan bobot yang didapatkan dari algoritma *multiobjective evolutionary* NSGA – II.
- e. Perhitungan nilai *return* portofolio dan risiko menggunakan *variance*.
- f. Perhitungan nilai *return* dan risiko menggunakan *semivariance*.
- g. Nilai risiko yang telah didapatkan menggunakan *semivariance* dan *variance* selanjutnya dibandingkan mana yang menghasilkan risiko minimum.
- h. Proses selesai.

3.2 Gambaran sistem

Berikut adalah gambaran dari Algoritma *Multiobjective Evolutionary* NSGA – II (*Non-dominated sort genetic algorithm*)



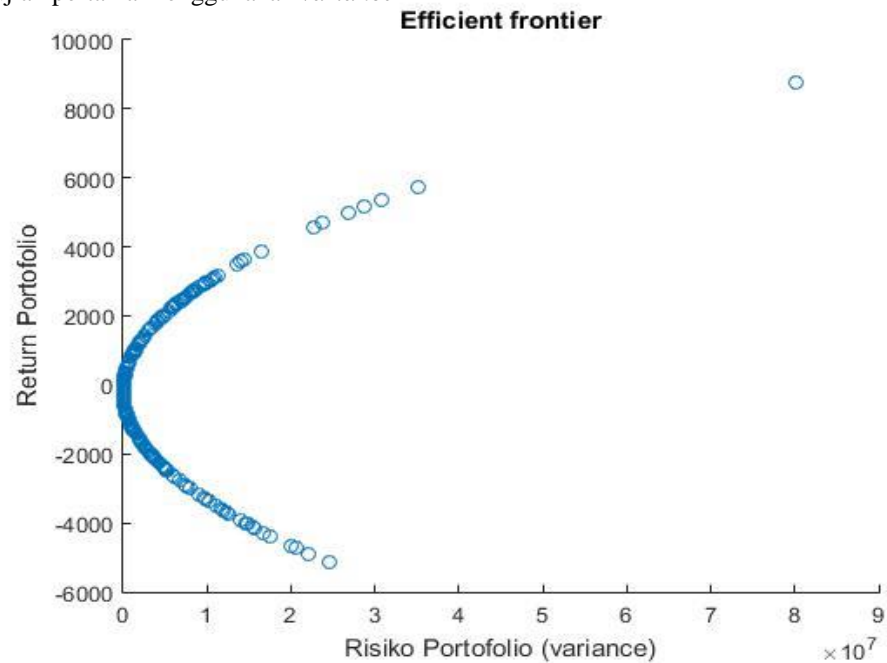
Gambar 3.2 Perancangan sistem NSGA – II

- a. *Start*
Langkah pertama yaitu input data harga saham penutupan *index* LQ45.
- b. *Inisialisasi Populasi*
Membangkitkan secara acak populasi yang berisikan beberapa kromosom. Setiap kromosom mengandung kombinasi bobot portofolio yang bernilai *real*.
- c. *Non-dominated Sorting*
Setelah dilakukan inisialisasi populasi, maka dilakukan *non-dominated sorting* dengan cara menentukan Individu yang termasuk *dominance* atau *non-dominance*, apabila tergolong individu *dominance* populasi terklasifikasi ke dalam *front-front*.
- d. *Fitness*
Individu yang tidak didominasi oleh individu yang lain (*non dominance*) kemudian dihitung nilai fitnessnya menggunakan 2 *objective*.
- e. *Crowding distance*
Crowding distance dilakukan untuk mempertahankan keragaman populasi, perhitungan dilakukan hanya pada *front* yang tidak mendominasi.
- f. *Binary Tournament Selection*
Ada 2 kriteria dalam seleksi orang tua menggunakan *binary tournament selection* yaitu : jika solusi menggunakan *non domination rank* hasil nilainya berbeda maka solusi dengan nilai *rank* yang terkecil yang dipilih sedangkan jika didapatkan solusi menggunakan *crowding distance* solusi dengan nilai terbesar yang dipilih.
- g. *Crossover dan Mutasi*
Operator *crossover* atau kawin silang digunakan untuk menghasilkan individu baru yang mewarisi sifat dari kromosom orang tua. Langkah selanjutnya adalah mutasi menggunakan *swap mutation*. Cara kerjanya adalah dengan menukar nilai gen 1 dengan gen lainnya dengan tujuan nilai bobot yang dihasilkan tetap 1.
- h. *Populasi Baru*
Populasi baru berisikan individu-individu hasil dari proses *crossover* dan mutasi.
- i. *Bobot*
Setelah sistem algoritma genetika *mutlobjective evolutionary NSGA – II* berhenti dijalankan kemudian diperoleh nilai berupa bobot yang akan digunakan untuk menghitung nilai *return* dan risiko portofolio yang optimal.

4. Evaluasi

Pada pengujian pertama pembentukan portofolio dilakukan menggunakan saham (ADHI, ASII, BBCA, BBNI, BBRI) dengan jumlah record 261 data. Setelah melakukan pengujian maka diperoleh hasil sebagai berikut :

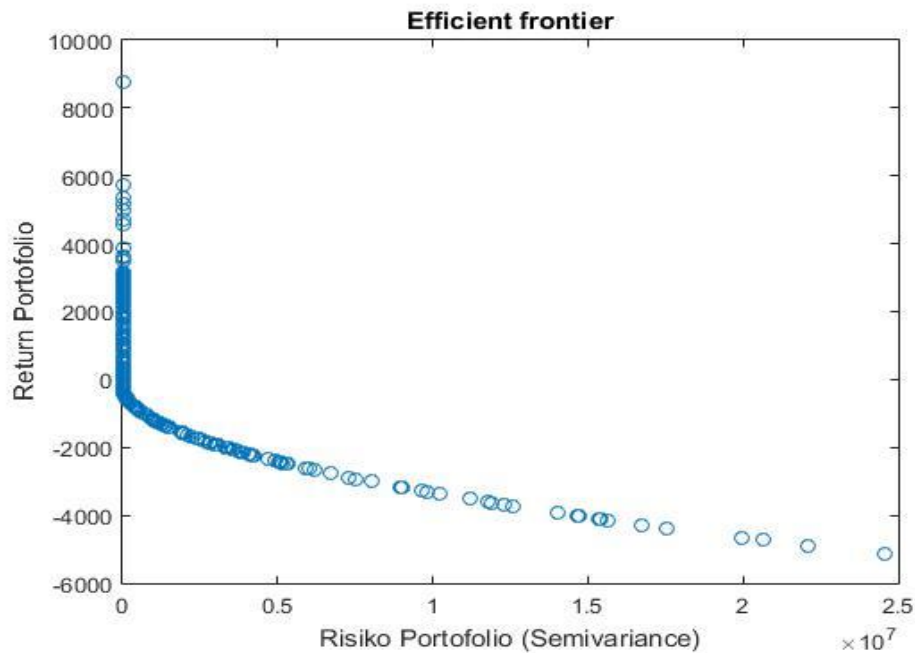
4.1 Hasil Pengujian pertama menggunakan *variance*



Gambar 3.1 Grafik return portofolio terhadap *variance*

Dari pengujian yang telah dilakukan didapatkan portofolio *multiobjective* menggunakan *variance*. Berdasarkan Gambar 3.1 diperoleh solusi - solusi return portofolio yang optimal. Portofolio yang optimal terlihat dari titik-titik yang nilai *return*-nya berada di atas nilai 0 dan risikonya mendekati nilai 0.

4.1 Hasil pengujian menggunakan *semivariance*



Gambar 3.2 Grafik portofolio menggunakan *semivariance*

Setelah dilakukan pengujian didapatkan solusi-solusi yang optimum. Berdasarkan Gambar 3.2 dapat disimpulkan bahwa solusi – solusi yang optimal diperoleh dari titik-titik yang nilai *return*-nya berada di atas nilai 0 dan risikonya mendekati nilai 0.

5. Kesimpulan

Berdasarkan hasil analisis terhadap implementasi dan pengujian sistem Optimasi Portofolio *Mean-semivariance* Algoritma Genetika *Multiobjective* NSGA-II dengan metode pembandingan *Mean variance*, maka dapat ditarik kesimpulan bahwa *semivariance* memiliki solusi yang lebih optimal karena seluruh himpunan solusi pada *semivariance* nilai return portofolio yang bernilai positif memiliki nilai risiko yang mendekati 0, sedangkan untuk himpunan solusi menggunakan *variance* jika return portofolio semakin tinggi nilai return portofolio, maka nilai risikonya juga semakin besar.

Daftar Pustaka

- [1] Luis Lobato Macedo, Pedro Godinho, Maria Joao Alves.(2017). Mean-Semivariance Portofolio Optimization with Multiobjective Evolutionary Algorithms and Technical Analysis Rules. Portugal:*ScienceDirect*
- [2] Javier Estrada.(2007). Mean-Semivariance Optimization: A Heuristic Approach. Barcelona, Spain: *IESE Business School*.
- [3] Capinski, M., & Zastawniak, T. (2003). *Mathematics for Finance : An Introduction fo Financial Engineering*. London: Springer.
- [4] E.Zitzler.Evolutionary Algorithm for Multiobjective Optimization : Method and Application.PHD Teshis,ETH Zurich,Switzerland,1999
- [5] Fifin Sonata, Muhammad Zarlis, Tulus (2016). Optimasi Makespan dan Total Tardiness Dalam Penjadwalan Mesin Produkti Type Flow Shop Menggunakan metode Non-Dominated Genetic Algorithm (NSGA II). *Jurnal Teknologi Informasi dan Komunikasi Vol 5 No 1, 1-12*
- [6] Deb, K, Pratap, A, Argawal, S, & Meyarivan, T, 2002, A fast elitist multiobjective genetic algorithm: NSGA – II , *IEEE Trans Evol Comput*, 182-97.
- [7] Sherly Isnaeni, Deni Saepudin, Rian Febrian Umbara, (2015). Penerapan Algoritma Genetika Multi-objective NSGA-II Pada Optimasi Portofolio Saham. *e-Proceeding of Engineering*, 6841
- [8] Yayat Priyatna, F. Sukono, (2003). Optimasi Portofolio Investasi Dengan Menggunakan Model Markowitz. *Jurnal Matematika Dan Komputer Vol. 6, No. 1, 1 - 10, April 2003, ISSN : 1410-8518*