

Optimisasi merupakan suatu upaya sistematis untuk memilih elemen terbaik dari suatu kumpulan elemen yang ada. Di dalam konteks matematika, optimisasi ini bisa dinyatakan sebagai suatu usaha sistematis untuk mencari nilai minimum atau maksimum dari suatu fungsi. Dengan kata lain, optimisasi merupakan proses mencari nilai terbaik berdasarkan fungsi tujuan dengan daerah asal yang telah didefinisikan. Fungsi ini secara sederhana dapat dinyatakan dengan model matematika seperti di bawah ini: fea

$$F_{opt}(x) = \begin{cases} \max(f(x)), & \text{goal} = \text{benefit} \\ \min(f(x)), & \text{goal} = \text{cost} \end{cases}$$

sebagai contoh adalah fungsi kuadrat  $f(x) = x^2$  dimana  $x \in \mathbb{R}$ . Di dalam contoh ini,  $f(x) = x^2$  merupakan fungsi tujuannya, sedangkan  $x$  adalah daerah asal yang didefinisikan sebagai anggota bilangan ril.

Konsep optimisasi sudah dipakai sejak jaman prasejarah. Hal ini dapat dibuktikan dengan adanya saluran-saluran air yang ditemukan di situs-situs presejarah. Saluran-saluran air ini dipakai untuk mengoptimalkan penggunaan air. Hal ini mengindikasikan bahwa konsep optimisasi merupakan bagian dari kehidupan manusia sejak lama. Permasalahan pengaturan air masih dijumpai dalam masyarakat masa kini, hanya saja penyelesaiannya sudah menggunakan metode optimisasi yang modern.

Meskipun konsep optimisasi sudah sangat lama digunakan, tetapi metode optimisasi pertama, yang mengacu pada teknik yang terstruktur, yang diakui adalah **Steepest Descent**. Istilah optimisasi diperkenalkan oleh **George Dantzig** yang mengembangkan **algoritma simplex** untuk menyelesaikan masalah linear programming. Istilah programming disini tidak mengacu kepada pemrograman komputer, tetapi lebih pada program pelatihan dan penjadwalan logistik yang diadakan oleh pihak militer Amerika dimana masalah-masalah tersebut menjadi focus riset yang dilakukan oleh Dantzig. **Linear programming** sendiri merupakan metode untuk menyelesaikan fungsi linear, baik fungsi tujuan maupun fungsi batasannya (*constraint*).

Dalam perkembangan selanjutnya, optimisasi sangat berkaitan dengan perkembangan komputer karena proses optimisasi ini umumnya dijalankan di komputer. Di awal-awal perkembangannya, penelitian optimisasi hanya dilakukan secara intensif untuk menyelesaikan permasalahan-permasalahan penting di bidang militer yang melibatkan teknologi tinggi. Tetapi

ketika harga komputer semakin terjangkau, penelitian di bidang optimisasi berkembang sangat pesat. Dalam dua dekade terakhir, banyak sekali metode optimisasi baru yang lahir.

Optimisasi dipakai di hampir semua bidang ilmu antara lain bidang teknik, sains, ilmu sosial, ekonomi dan bisnis. Banyak permasalahan di bidang teknik, sains dan ekonomi yang dapat dinyatakan sebagai permasalahan optimisasi seperti meminimalkan biaya, waktu dan resiko atau memaksimalkan keuntungan dan kualitas. Optimisasi seringkali menjadi focus utama dalam pengambilan keputusan misalnya untuk meningkatkan daya saing suatu produk, maka perusahaan harus bisa memaksimalkan kualitas dari produk tersebut dengan meminimalkan biaya produksi. Pengambilan keputusan terdiri dari beberapa langkah (Talbi, 2009):

1. Merumuskan masalah

Pada langkah ini, dilakukan proses identifikasi permasalahan yang akan diselesaikan, seperti menentukan faktor-faktor yang mempengaruhi permasalahan, lalu menentukan fungsi tujuan dari permasalahan tersebut.

2. Memodelkan masalah

Pada langkah ini, dilakukan proses pemodelan dari permasalahan yang akan diselesaikan, permasalahan yang sesungguhnya bisa jadi sangat kompleks serta sulit untuk dideskripsikan, sehingga perlu adanya penyederhanaan supaya bisa dinyatakan secara matematis.

3. Optimalisasi

Setelah permasalahan dimodelkan secara matematis, metode optimisasi diterapkan untuk mendapatkan penyelesaian yang baik. Seringkali model permasalahan yang dibuat tidak bisa mewakili permasalahan sesungguhnya, sehingga kepresisian dalam memodelkan permasalahan sangat mempengaruhi hasil yang didapatkan. Penyelesaian yang didapatkan bisa merupakan solusi optimal maupun non-optimal.

4. Menerapkan penyelesaian masalah

Penyelesaian yang didapatkan kemudian diterapkan untuk mengetahui apakah penyelesaian tersebut dapat diterima atau tidak. Jika penyelesaian tersebut tidak bisa diterima, maka model permasalahan maupun metode optimisasinya perlu diperbaiki dan prosesnya diulang lagi.

Kata Kunci: Komputasi, Optimisasi

Permasalahan optimisasi bisa dibedakan menjadi permasalahan diskrit dan kontinu. Permasalahan dikatakan **diskrit** jika variabel keputusannya bernilai diskrit. Permasalahan diskrit memiliki kandidat penyelesaian yang terbatas tetapi ukurannya meningkat sangat pesat sesuai dengan dimensi permasalahannya (faktorial). Hal ini menyebabkan metode enumerasi tidak memungkinkan untuk dilakukan, karena memerlukan waktu yang sangat lama untuk mendapatkan penyelesaian. Contoh yang cukup populer dari permasalahan optimisasi diskrit adalah **Traveling Salesman Problem**. Berbeda dengan permasalahan diskrit, permasalahan **kontinu** memiliki variabel yang bernilai kontinu. Permasalahan ini memiliki kandidat penyelesaian yang tak terbatas. Umumnya batas-batas ruang pencarian dalam permasalahan kontinu sudah diketahui, sehingga pencarian hanya dibatasi pada ruang tersebut.

Berdasarkan jumlah tujuan dari permasalahan yang akan dioptimisasi, model optimisasi dapat dibedakan menjadi single-objective dan multi-objective. Optimisasi **single-objective** berarti hanya memiliki satu tujuan, misalnya meminimalkan biaya, sedangkan **multi-objective** memiliki beberapa tujuan, misalnya meminimalkan biaya dan waktu sekaligus.

Model penyelesaian permasalahan optimisasi dapat dikelompokkan menjadi penyelesaian **eksak** dan juga pendekatan atau **aproksimasi** (*approximation*).

Kata Kunci: Exact Method, Komputasi, Metaheuristik, Optimisasi

Heuristik berasal dari kata Yunani ‘heuriskein’ yang berarti seni untuk menemukan strategi dalam menyelesaikan persoalan sedangkan ‘meta’ berarti metodologi tingkat tinggi atau lanjut (Talbi, 2009). Di dalam ilmu komputer, metode heuristik merupakan suatu teknik untuk penyelesaian permasalahan yang tidak menekankan pada pembuktian, apakah solusi yang didapatkan adalah benar, tetapi lebih menekankan pada performa komputasi dan kesederhanaan. Pembuktian apakah suatu solusi adalah benar merupakan fokus dari metode penyelesaian analitik. Metode heuristik merupakan suatu metode penyelesaian yang menggunakan konsep pendekatan atau aproksimasi (*approximation*).

Menurut Talbi (2009), metaheuristik dapat didefinisikan sebagai metode lanjut (*advanced*) berbasis heuristik untuk menyelesaikan persoalan optimisasi secara efisien. Pada wikipedia, metaheuristik didefinisikan sebagai metode optimisasi yang dilakukan dengan memperbaiki kandidat penyelesaian secara iteratif sesuai dengan fungsi objektifnya. Metode ini mampu menghasilkan penyelesaian yang baik dalam waktu yang cepat (*acceptable*), tetapi tidak menjamin bahwa penyelesaian yang dihasilkan merupakan penyelesaian terbaik (*optimal*). Metode metaheuristik banyak dipakai dalam optimisasi stokastik, yaitu optimisasi yang memiliki derajat ketidakpastian atau *random*.

Menurut Blum & Roli (2003), metaheuristik memiliki beberapa karakteristik dasar yaitu:

- a) Metaheuristik adalah strategi yang memandu proses pencarian;
- b) Tujuan dari metaheuristik adalah untuk menjelajahi ruang pencarian secara efisien untuk menemukan solusi optimal;
- c) Teknik metaheuristik berkisar dari prosedur pencarian lokal yang sederhana sampai proses pembelajaran yang kompleks;
- d) Metaheuristik adalah metode pendekatan dan biasanya non-deterministik;
- e) Metaheuristik dapat terdiri dari penggabungan beberapa mekanisme supaya proses pencarian tidak terjebak dalam daerah terbatas di ruang pencarian;
- f) Konsep dasar dari metaheuristik memungkinkan pendeskripsian secara abstrak;
- g) Metaheuristik bersifat *general* atau umum sehingga dapat diterapkan dalam berbagai macam persoalan;
- h) Metaheuristik dapat menggunakan domain pengetahuan khusus dalam bentuk heuristik yang dikendalikan dengan strategi tingkat lanjut;

- i) Metaheuristik dapat menggunakan pengalaman yang didapat selama proses pencarian untuk menuntun proses pencarian;

Dalam menentukan apakah metaheuristik adalah metode yang sesuai untuk menyelesaikan suatu permasalahan, ada beberapa hal yang perlu diperhatikan misalnya kompleksitas permasalahan, ukuran input, struktur input, dan waktu yang diperlukan untuk menyelesaikan masalah tersebut. Secara umum metaheuristik dipakai untuk masalah-masalah yang kompleks dan tidak bisa diselesaikan dengan mudah secara analitikal atau eksak. Tidaklah terlalu bermanfaat menggunakan metaheuristik untuk persoalan yang dengan mudah dan cepat dapat diselesaikan secara eksak. Penyelesaian eksak merupakan penyelesaian terbaik, tetapi seringkali metode ini tidak dapat diterapkan pada permasalahan optimisasi, sehingga dipakailah metode pendekatan.

Metode heuristik lebih bersifat 'problem spesifik' sehingga hanya diterapkan untuk jenis permasalahan tertentu. Sedangkan metaheuristik lebih bersifat umum, sehingga dapat dipakai untuk berbagai macam jenis permasalahan optimisasi. Kata 'meta' sendiri artinya adalah 'higher level'. Biasanya, metode heuristik diadopsi di dalam metode metaheuristik, untuk meningkatkan performa dari metaheuristik.

**NP** merupakan jenis permasalahan optimisasi yang penyelesaiannya dapat diverifikasi atau dibuktikan berada dalam batasan waktu polinomial  $O(n^k)$ , dimana  $n$  merupakan ukuran permasalahannya dan  $k$  adalah suatu konstanta. Di dalam **NP**, ada permasalahan yang sudah bisa diselesaikan dalam batasan waktu polinomial, dimana algoritma untuk menyelesaikan persoalan tersebut dalam batasan waktu polinomial sudah diketahui, kemudian ada juga permasalahan yang algoritma untuk menyelesaikannya dalam batasan waktu polinomial belum diketahui, tetapi juga tidak bisa dibuktikan bahwa algoritma tersebut tidak ada.

**NP-hard** secara informal didefinisikan sebagai permasalahan optimisasi yang setidaknya sama sulitnya dengan permasalahan **NP** yang tersulit. Maksudnya adalah sebagai berikut:

- a) Algoritma yang bisa menyelesaikan permasalahan tersebut dalam batasan waktu polinomial belum diketahui;
- b) Di dalam **NP-hard**, ada permasalahan yang sudah bisa dibuktikan dapat diselesaikan dalam batasan waktu polinomial dan ada juga yang belum bisa atau kemungkinan tidak bisa dibuktikan dapat diselesaikan dalam batasan waktu polinomial;

Contoh permasalahan **NP-hard**: Traveling Salesman Problem

**NP-complete** adalah permasalahan yang masuk kategori *NP* sekaligus *NP-hard*, yaitu irisan antara *NP* dan juga *NP-hard*. Maksudnya adalah sebagai berikut:

- a) Penyelesaian permasalahan *NP-complete* dapat diverifikasi atau dibuktikan berada dalam batasan waktu polinomial;
- b) Algoritma yang bisa menyelesaikan permasalahan tersebut dalam batasan waktu polinomial belum diketahui, tetapi juga tidak bisa dibuktikan bahwa algoritma tersebut tidak ada.

Metode analitikal adalah metode yang pasti menghasilkan penyelesaian optimal. Analitikal maksudnya adalah jawaban yang diberikan dapat dibuktikan secara matematis sebagai jawaban yang optimal. Metode analitikal seringkali disebut juga dengan 'exact method'. Salah satu contohnya yaitu linear programming.

### **Tambahan**

Algoritma optimasi merupakan suatu metode untuk menyelesaikan permasalahan optimisasi. Algoritma ini dibagi menjadi dua, yaitu metode eksak atau exact method / metode analitikal dan juga metode pendekatan atau approximate method.

Exact method menjamin bahwa hasil penyelesaian yang diperoleh adalah optimal, sedangkan approximate method tidak menjamin bahwa hasil yang diperoleh adalah hasil optimal, meskipun tidak optimal, tetapi memiliki kualitas yang baik.

Metode heuristik dan juga metaheuristik termasuk approximate method, tetapi metaheuristik bisa dipakai untuk berbagai jenis permasalahan optimisasi. Metode metaheuristik merupakan metode optimisasi general purpose yang didasarkan pada metode heuristik.

Perbedaan utama heuristik dan metaheuristik yaitu, metode heuristik merupakan problem dependent, yang artinya metode heuristik bergantung pada jenis permasalahan atau hanya bisa digunakan untuk jenis permasalahan tertentu, sedangkan metode metaheuristik merupakan problem independent, yang artinya metode metaheuristik bisa digunakan untuk berbagai jenis permasalahan.

Baik heuristik maupun metaheuristik keduanya adalah jenis algoritma optimisasi. **Ant Colony Optimization** (ACO) termasuk metode metaheuristik, lebih spesifiknya population-based metaheuristics dan masuk kategori swarm intelligence algorithm.

Dalam permasalahan optimisasi, istilah solusi terbaik sering disamakan dengan solusi optimal. Solusi terbaik kadang masih bersifat relatif, artinya terbaik dari yang ada / yang bisa didapat. Kalau pengertiannya seperti itu, metode metaheuristik dapat menghasilkan solusi terbaik dari alternatif solusi yang ada. Tapi kalau optimal, pengertiannya global, artinya terbaik dari semua kemungkinan penyelesaian yang ada di dalam ruang pencarian.

Secara teknis sebenarnya metode dan algoritma tidak memiliki perbedaan yang jelas dan sering dipakai secara bergantian, akan tetapi menurut pemahaman saya, metode merupakan prosedur yang bersifat umum sedangkan algoritma sering dikaitkan pada suatu prosedur spesifik untuk menyelesaikan permasalahan. Metode, menurut saya tidak harus dipakai untuk menyelesaikan masalah, misalnya metode membaca cepat, tetapi kalau algoritma, selalu dipakai untuk menyelesaikan masalah.

Untuk istilah metaheuristik, saya lebih suka menyebutnya metode, karena metaheuristik ini lebih ke tataran konseptual saja. Dan metode metaheuristik terdiri dari banyak jenis metode yang lebih spesifik, yang orang sering menyebutnya dengan istilah algoritma, misalnya algoritma genetik, algoritma differential evolution, dll.

Kata Kunci: Komputasi, Metaheuristik, Optimasi

Alam merupakan sumber inspirasi yang tiada habis-habisnya. Beberapa fenomena alam memiliki sifat ‘cerdas’. Dalam sudut pandang optimisasi, terdapat 3 (tiga) fenomena alam yang banyak digunakan sebagai analogi di dalam optimisasi, yaitu *evolution theory*, *self-organized criticality*, dan *emergence theory*.

### 1. Evolution Theory

Teori evolusi erat kaitannya dengan **Charles Darwin** yang pertama kali memformalisasikan teori tersebut dalam konteks teori biologi. Teori ini telah mengundang banyak perdebatan bahkan sampai sekarang, khususnya yang berkaitan dengan asal-usul makhluk hidup. Lalu bagaimana pengaruh teori evolusi dalam bidang optimisasi?

Teori evolusi bisa dipandang proses optimisasi yang didasarkan pada konsep kompetisi. Di dalam evolusi, ada dua mekanisme yang penting yaitu pewarisan sifat (*inheritance*) dan mutasi. *Inheritance* memegang peranan penting karena dapat menurunkan sifat-sifat, baik dari sang induk atau moyang sedangkan mutasi merupakan suatu proses perubahan pada individu yang tidak dipengaruhi oleh individu lainnya. Bisa jadi, mutasi ini adalah suatu mekanisme untuk beradaptasi dengan perubahan lingkungannya. Di dalam evolusi, makhluk hidup (*species*) yang mampu beradaptasi dengan lingkungannya, akan memiliki peluang untuk bertahan hidup lebih lama sedangkan yang tidak mampu beradaptasi akan segera punah.

Konsep dari evolusi ini diadopsi menjadi sebuah mekanisme untuk mencari penyelesaian dari masalah optimisasi yang sering disebut dengan **evolutionary algorithm**. Metode ini memakai banyak konsep yang meminjam dari proses evolusi biologis seperti seleksi, mutasi, crossover, kromosom, dan lain-lain. Kualitas dari suatu individu dinyatakan dengan nilai *fitness*.

Kromosom merupakan istilah yang menyatakan representasi dari suatu permasalahan. Seleksi berfungsi untuk memilih individu yang akan dijadikan sebagai induk. Mutasi dan crossover merupakan mekanisme untuk rekombinasi dalam rangka menghasilkan anak (*offspring*).

Ada beberapa macam pengelompokan dari *evolution algorithm*. Akan tetapi, banyak ahli yang mengklasifikasikan *evolution algorithm* menjadi 4 sub-kelas, diantaranya: (a) **Algoritma genetik**; (b) **Evolution strategies**; (c) **Genetic programming**; (d) **Evolution programming**.



## 2. Self-Organized Criticality

Banyak fenomena alam yang menunjukkan adanya sifat ‘self organizing’ atau sebuah sifat yang membuat sesuatu seperti mengatur dirinya sendiri. Fenomena ini banyak sekali, misalnya, ambillah seember pasir kemudian tuangkan secara perlahan-lahan ke tanah. Pasir tersebut akan secara alami ‘mengatur posisinya’ sehingga membentuk sebuah gundukan berbentuk kerucut. Pasir-pasir tersebut seolah-olah memiliki kemampuan untuk mengatur dirinya sendiri menjadi sebuah gundukan berbentuk kerucut dan bukan menyebar seperti air.

Self-organized critically merupakan sebuah sifat dari sistem dinamis yang mana memiliki sebuah titik kritis atau disebut juga dengan **attractor**. Adanya titik kritis inilah yang menyebabkan perilaku self-organized, tanpa perlu adanya sebuah parameter kontrol dari luar.

Self-organized critically diperkenalkan oleh **Bak, Tang, & Wiesenfeld** pada tahun 1988. Mereka menjabarkan self-organized critically sebagai sebuah karakteristik kompleks suatu sistem yang dinamis. Di dalam artikel tersebut, mereka mempelajari karakteristik sistem dinamis yang memiliki derajat kebebasan temporal dan spatial. Dalam paper tersebut, mereka menyatakan bahwa sebuah sistem dinamis dengan derajat kebebasan yang telah di perbesar (*extends*) dapat berkembang menuju kondisi kritis yang menyebabkannya menjadi self-organized.

Konsep self-organized critically ini banyak dipadukan dengan metode evolution algorithm, misalnya algoritma genetik, untuk menyelesaikan permasalahan optimisasi.

## 3. Emergence Theory

Sudah sejak jaman dahulu, manusia mengamati bahwa mekanisme yang kompleks dapat dihasilkan dari kerja sama antara agent-agent yang memiliki prosedur sederhana. Sebagai contoh adalah kinerja dari semut. Semut dapat membangun sarangnya, menemukan makanan, dan mempertahankan koloninya melalui kerja sama antar semut. Tiap semut sendiri memiliki kerja yang sederhana, tetapi ketika sudah menjadi sebuah koloni, maka dapat mengerjakan hal-hal yang menakjubkan.

Konsep ini yang mendorong lahirnya beberapa metode seperti **agent-based system** maupun **swarm intelligence**. Di dalam bidang optimisasi, swarm intelligence telah melahirkan metode-metode yang cukup fenomenal seperti **Particle Swarm Optimization**, **Bee Colony Optimization** dan **Ant Colony Optimization**. Metode ini merupakan sebuah metode terdistribusi, dimana setiap agent akan bekerja dengan sejumlah aturan yang sederhana. Yang membuatnya menjadi

begitu handal adalah, agent tersebut memiliki mekanisme komunikasi yang memungkinkan adanya kerja sama antar agent.

Metode ini sangat sesuai dalam menyelesaikan permasalahan optimisasi yang gambaran keseluruhannya susah untuk didapatkan. Misalnya untuk mengoptimalkan proses routing pada jaringan komputer.

Kata Kunci: Komputasi

Teori ini diperkenalkan oleh **David Wolpert** dan **William G. Macready** tahun 1995. Secara singkat, teori ini memberikan implikasi terhadap dua hal.

Yang pertama, teori ini menyatakan bahwa jika suatu algoritma memiliki performa yang baik di suatu jenis permasalahan optimisasi, maka algoritma itu akan memiliki performa yang tidak baik di jenis permasalahan lainnya. Dan semua algoritma itu pada dasarnya memiliki performa yang ‘sama’ atau sebanding jika dihitung rata-rata berdasarkan pada semua kemungkinan permasalahan yang ada.

Hal ini mengindikasikan bahwa secara teoritis, tidaklah mungkin membuat satu algoritma yang bisa menyelesaikan semua permasalahan dengan sangat memuaskan. Setiap metode hanya cocok untuk suatu jenis permasalahan tertentu. Misalnya, konsep **nearest neighbor** sesuai untuk permasalahan sejenis travelling salesman problem, konsep **ant colony** sangat sesuai diterapkan pada permasalahan dynamic routing, dll.

Efek lainnya, karena jumlah permasalahan optimisasi semakin bertambah, maka metode-metode optimisasi baru sangat diperlukan. Sederhananya, karena satu metode hanya cocok untuk satu jenis permasalahan, maka semakin banyak permasalahan akan membutuhkan semakin banyak metode penyelesaian. Misalnya, ditemukannya komputer memunculkan masalah di bidang routing jaringan. Adanya internet memunculkan permasalahan di bidang pencarian informasi secara cepat. Hal-hal seperti ini menjadi ladang yang subur sekaligus tantangan buat pencipta algoritma. Dan temuan-temuan baru di bidang algoritma sangatlah banyak.

Implikasi kedua, efektivitas suatu metode sering kali berbanding terbalik dengan efisiensi-nya. Jika suatu metode memiliki efektivitas yang sangat bagus dalam menyelesaikan masalah dengan tepat, seringkali diperoleh dengan mengorbankan efisiensinya, seperti memerlukan komputasi yang lama ataupun lebih lambat.

Akibatnya, menentukan jenis metode yang tepat untuk menyelesaikan suatu permasalahan juga tidaklah mudah. Hal itu tergantung dari seberapa presisi / tepat solusi yang diinginkan, dan berapa waktu yang disediakan. Sebagai contoh, perhitungan dalam mendesain badan pesawat luar angkasa sangatlah penting bagi keselamatan, sehingga kepresisian tinggi sangatlah diperlukan. Untuk itu, proses perhitungan yang memerlukan waktu beberapa hari atau minggu tidaklah menjadi masalah. Disisi lain, metode untuk mendeteksi wajah dalam

kamera digital tidak mentolerir waktu yang lama, karena pengguna kamera tidak akan mau menunggu waktu berjam-jam hanya untuk hal tersebut. Sehingga, kecepatan menjadi prioritas utama dan tidak mengherankan jika seringkali program di kamera itu tidak mampu mendeteksi wajah dengan baik.

Terlepas dari permasalahan di dunia optimisasi, teori 'no free lunch' ini sangat mudah kita jumpai dalam kehidupan sehari-hari. Misalnya, jika kita ingin dapat nilai yang baik, maka harga yang harus dibayar adalah belajar dengan tekun. Jika ingin menjadi atlet yang tangguh, maka harga yang harus dibayar adalah berlatih dengan tekun.

Kata Kunci: Komputasi, Optimasi

Metode optimisasi dapat dibagi menjadi dua kategori, yaitu metode eksak (*analitical*) dan metode pendekatan (*approximate*).

Ciri yang paling khas dari metode eksak adalah bahwa metode ini akan menghasilkan penyelesaian yang optimal. Bagaimana bisa tahu bahwa hasil yang diperoleh adalah hasil optimal? Pembuktiannya dilakukan secara analitis menggunakan metode matematis.

Sedangkan metode pendekatan tidak menjamin bahwa hasil penyelesaiannya adalah yang optimal. Tetapi, meskipun metode ini tidak menjamin hasil optimal, tetapi umumnya hasil penyelesaiannya cukup baik. Walaupun definisi baik ini juga sulit, karena bersifat kualitatif. Sehingga seringkali untuk mengukur kualitas baik ini, maka diadakan perbandingan hasil dengan menggunakan metode lain. Suatu metode dikatakan lebih baik jika dia menghasilkan penyelesaian yang lebih bagus daripada metode lainnya.

Metode pendekatan ada bermacam-macam, contohnya metode heuristik dan metaheuristik. Metaheuristik pada dasarnya merupakan metode pendekatan yang didasarkan pada metode heuristik. Sehingga tidak heran bahwa metode heuristik sering kali diintegrasikan di dalam metode metaheuristik.

Perbedaan utama dari metode heuristik dan metaheuristik, yaitu metode heuristik bersifat problem dependent sedangkan metode metaheuristik bersifat problem independent. Lalu apakah yang dimaksud dengan problem dependent dan problem independent itu?

Problem dependent artinya bergantung pada permasalahan, jadi metode heuristik itu hanya bisa dipakai untuk jenis permasalahan tertentu. Misalnya, metode nearest neighborhood termasuk metode heuristik. Metode tersebut hanya bisa dipakai pada permasalahan yang mengenal konsep neighborhood / tetangga, misalnya pada Traveling Salesman Problem maupaun Vehicle Routing Problems. Contoh lainnya adalah, COMSOAL (Computer Method for Sequencing Operations for Assembly Lines) hanya dipakai untuk persoalan sequencing assembly line.

Sedangkan problem independent berarti tidak bergantung pada jenis permasalahan. Jadi penerapan metode metaheuristik tidak bergantung pada jenis permasalahan, alias bisa dipakai untuk berbagai jenis permasalahan. Contoh dari metode metaheuristik adalah algoritma genetik, particle swarm optimization, ant colony optimization, dan lain sebagainya. Meskipun bisa dipakai untuk berbagai jenis permasalahan, tetapi kemampuan mengadopsi metode tersebut untuk jenis permasalahan tertentu berpengaruh besar terhadap kualitas penyelesaian

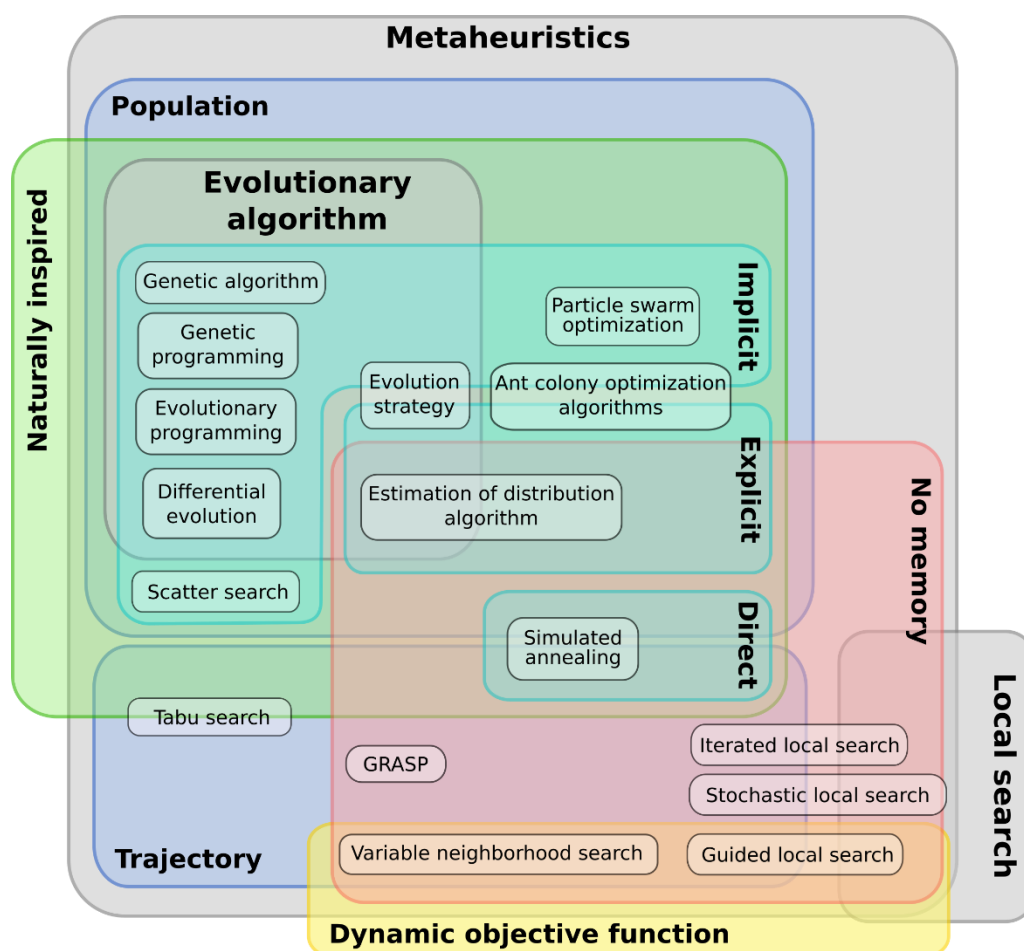
yang dihasilkan. Oleh karena itulah, seringkali metode metaheuristik akan mengintegrasikan metode heuristik dalam implementasinya. Misalnya, untuk menyelesaikan persoalan Traveling Salesman Problem, metode algoritma genetik menyisipkan konsep nearest neighborhood di dalam implementasinya.

Kata Kunci: Komputasi, Optimasi

Metode metaheuristik dapat diklasifikasikan menjadi berbagai macam kriteria, misalnya:

- a) Berdasarkan jumlah solusi: solusi tunggal, populasi;
- b) Berdasarkan memori: memory-based, memory-less;
- c) Berdasarkan fokus pencarian: local search, global search;
- d) Berdasarkan inspirasi: nature, non-nature;

Salah satu model klasifikasi metaheuristik yang saya suka adalah model yang divisualkan oleh Johann Dréo. Bentuk visualnya menarik dan mudah dimengerti.



Swarm intelligence berada sejajar dengan evolutionary algorithm. Dan kedua kelas tersebut menjadi metode metaheuristik yang paling banyak diterapkan hingga saat ini.

Kata Kunci: Komputasi, Metaheuristik, Optimasi

Tabu search merupakan metode metaheuristik single-solution based yang diperkenalkan oleh **Fred Glover** pada tahun 1986. Tabu search sangat populer di tahun 90-an, dan sampai sekarang masih menjadi salah satu metode metaheuristik single-solution based yang banyak dipakai untuk menyelesaikan permasalahan optimisasi. Tabu search merupakan metode metaheuristik yang dilandaskan pada **local search**.

Tabu search mempunyai tiga komponen penting:

- a) Penggunaan struktur memori yang fleksibel, yang memungkinkan evaluasi kriteria dan informasi historis bisa dieksploitasi lebih baik dibandingkan dengan struktur memori yang statis.
- b) Penggunaan mekanisme kendali yang didasarkan pada interaksi antara kondisi yang membatasi dan kondisi yang mendukung proses pencarian.
- c) Penggabungan fungsi memori yang memiliki rentang waktu yang berbeda, dari memori jangka pendek sampai dengan memori jangka panjang, untuk menerapkan strategi intensifikasi dan diversifikasi.

Untuk menghindari proses pencarian kembali ke kandidat solusi yang pernah dikunjungi, tabu search akan mengingat jalur pencarian yang telah dilewati. Kandidat solusi yang sudah dilewati akan disimpan dalam memori, yang disebut tabu list, dan tidak akan dilewati lagi. Dalam setiap iterasi, memori ini akan diperbarui. Akan tetapi, menyimpan semua solusi yang pernah ditemukan membutuhkan waktu dan kapasitas memori yang besar, karena dalam setiap iterasi harus dilakukan pencocokkan apakah suatu kandidat solusi ada diantara tabu list, sehingga tabu list biasanya ditetapkan berukuran konstan dan berisi pergerakan yang tabu.

Dalam penerapan tabu search, ada beberapa hal yang harus diperhatikan, yaitu:

- a) Tabu list, bertujuan untuk menghindari evaluasi kandidat solusi yang pernah dikunjungi sebelumnya. Akan tetapi, menampung semua kandidat solusi yang pernah dievaluasi ke dalam tabu list akan membuat tabu search menjadi tidak efektif.
- b) Aspiration criteria, umumnya diterapkan jika pergerakan tabu tersebut menghasilkan kandidat solusi yang memiliki nilai yang lebih baik daripada solusi terbaik yang telah dihasilkan.



- c) Intensifikasi, memori jangka menengah menyimpan sejumlah solusi yang berkualitas yang dihasilkan selama proses pencarian. Memori jangka menengah ini bertujuan untuk memberikan prioritas kepada atribut dari solusi berkualitas tersebut.
- d) Diversifikasi, memori jangka panjang menyimpan informasi kandidat solusi yang pernah dikunjungi. Berdasarkan informasi tersebut, memori ini dapat mengeksplorasi area dalam ruang pencarian yang belum dikunjungi.

Prosedur tabu search secara garis besar dapat dinyatakan sebagai berikut (Talbi, 2009):

```

bangkitkan solusi awal  $S(0)$ 
inisialisasi tabu list, intensifikasi memory, diversifikasi memory
while belum memenuhi kriteria berhenti
    tentukan solusi terbaik yang bisa diterima  $S'$ 
     $S = S'$ 
    update tabu list, aspiration, memory
    if memenuhi intensifikasi then intensifikasi
    if memenuhi diversifikasi then diversifikasi
endwhile
output: solusi terbaik yang ditemukan
  
```

Kata Kunci: Komputasi, Metaheuristik, Optimasi

Algoritma genetik merupakan salah satu metode metaheuristik yang paling populer. Hal ini disebabkan karena algoritma genetik memiliki performa yang baik untuk berbagai macam jenis permasalahan optimisasi. Algoritma genetik diperkenalkan oleh **Holland (1975)** dalam bukunya yang berjudul “Adaptation in Natural and Artificial Systems”. Adaptasi menjadi prinsip yang penting di dalam algoritma genetik. Adaptasi adalah kemampuan untuk menyesuaikan diri dengan lingkungannya, dan di dalam algoritma genetik, adaptasi dinyatakan dengan proses memodifikasi struktur individu yang akan meningkatkan kinerja algoritma genetik. Mekanisme kerja algoritma genetik mengikuti fenomena evolusi genetika yang terjadi dalam makhluk hidup.

Menurut **(Jacob, 2001)**, ada empat prinsip dasar di dalam algoritma genetik:

- a) Prinsip dualisme
- b) Pengkodean diskrit
- c) Efek rekombinasi
- d) Building blok dasar

Di dalam sistem biologi, **prinsip dualisme** yang dimaksudkan adalah informasi genetik yang terdapat di dalam DNA, selain memiliki fungsi sebagai informasi yang dapat direplikasi, juga berfungsi sebagai suatu instruksi yang harus dieksekusi. Pada algoritma genetik, struktur genetik dari individu dimodifikasi berdasarkan operasi rekombinasi dan mutasi, sedangkan instruksi genetik dinyatakan secara terpisah melalui berbagai parameter di dalam algoritma genetik. Pada awal perkembangannya, pengkodean di dalam algoritma genetik menggunakan pengkodean diskrit, terutama pengkodean **biner**. Di dalam pengkodean biner, setiap string memiliki nilai 1 atau 0. Selain pengkodean biner, pengkodean **permutasi** merupakan jenis pengkodean yang cukup populer, tetapi hanya bisa diterapkan pada permasalahan pengurutan seperti Travelling Salesman Problem. Di dalam pengkodean permutasi, angka di dalam string dinyatakan secara berurutan.

Kromosom	1	0	0	1	0	1	1	0	1
----------	---	---	---	---	---	---	---	---	---

a. Pengkodean biner

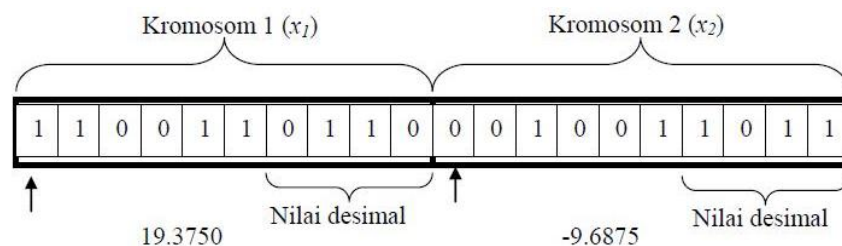
Kromosom	1	2	5	6	3	8	4	9	7
----------	---	---	---	---	---	---	---	---	---

b. Pengkodean permutasi

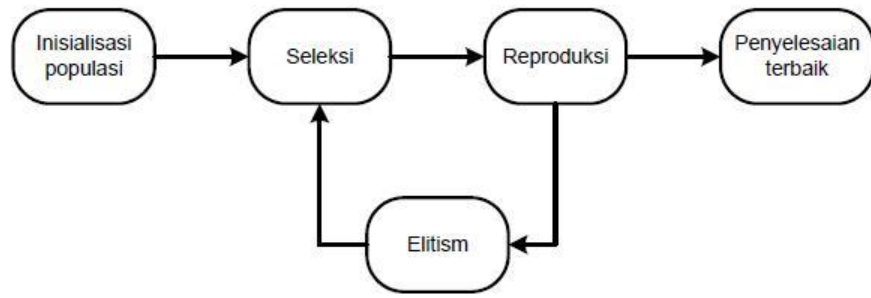
Algoritma genetik meniru proses rekombinasi dari sistem biologi melalui dua operator utamanya, crossover dan mutasi. **Crossover** akan meneruskan sifat-sifat baik yang terdapat pada induk kepada anaknya sedangkan **mutasi** berfungsi untuk menjaga keberagaman dari populasi. Representasi penyelesaian di dalam algoritma genetik dinyatakan dengan **kromosom**. Satu kromosom biasanya meyakakan satu buah variable penyelesaian dan setiap kromosom bisa terdiri dari beberapa gen. Sebagai contoh, fungsi sphere 2 dimensi sebagai berikut:

$$f_s = \sum_{i=1}^2 x_i^2$$

dengan pengkodean biner 10 bit, dimana digit paling kiri untuk menyatakan nilai positif dan negative, dan 4 digit paling kanan untuk menyatakan nilai pecahan. Maka, setiap representasi penyelesaian bisa dinyatakan sebagai individu dengan 2 kromosom, dimana tiap kromosom terdiri dari 10 gen.



Prosedur algoritma genetik diawali dengan proses inisialisasi populasi. Populasi terdiri dari sejumlah individu yang tersusun atas kromosom-kromosom. Individu yang telah dibangkitkan kemudian dievaluasi untuk menentukan nilai fitnessnya. Nilai fitness merupakan suatu nilai yang menyatakan kualitas dari individu, dan biasanya dirumuskan dengan fungsi objektif. Setelah evaluasi kemudian dilakukan seleksi untuk menentukan individu di dalam populasi yang akan dijadikan sebagai induk dalam proses reproduksi. Ada dua operator di dalam proses reproduksi yaitu crossover dan mutasi. Individu hasil reproduksi, yang disebut anak, kemudian dievaluasi. Apabila nilai fitness anak lebih baik daripada nilai fitness individu di dalam populasi, maka individu tersebut akan digantikan dengan anak. Proses penggantian ini sering disebut dengan elitism.



Kata Kunci: Komputasi, Metaheuristik