

**LAPORAN TUGAS PENDALAMAN MATERI**  
**MATA KULIAH KECERDASAN KOMPUTASIONAL**

**Dosen Pengampu : Dr. Wiharto, S.T., M.Kom.**



Disusun oleh:

**Kelompok 8**

M0521062 Putri Rizqi Khairunnisa

M0521071 Sekar Gesti Amalia Utami

M0521081 Yonaka Titin Nur Cahyani

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA**  
**UNIVERSITAS SEBELAS MARET**  
**TAHUN 2024**

## **PENDALAMAN MATERI**

### **1. Pada jenis permasalahan apa algoritma heuristik seharusnya diterapkan?**

**Jawab:**

Algoritma heuristik seharusnya diterapkan pada masalah optimasi yang kompleks atau masalah dengan ruang pencarian yang sangat besar, di mana solusi optimal sangat sulit ditemukan dengan metode konvensional. Heuristik cocok untuk kasus di mana ukuran data besar dan terdapat banyak variabel yang saling berinteraksi sehingga memerlukan waktu yang lama untuk menemukan solusi optimal. Contoh masalah yang sering diselesaikan dengan heuristik diantaranya:

- Pembuatan jadwal kuliah untuk menghindari bentrok antara ketersediaan dosen, ruangan, dan jadwal mahasiswa.
- Pendistribusian produk dari sumber ke tujuan dengan tujuan meminimalkan biaya.
- Pemilihan rute terpendek pada masalah Traveling Salesman Problem (TSP).
- Perancangan jaringan distribusi, seperti penempatan antenna dalam komunikasi. Heuristik umumnya digunakan untuk menemukan solusi mendekati optimal dalam waktu yang lebih cepat, meskipun tidak selalu memberikan hasil yang benar-benar optimal

### **2. Jelaskan apa yang dimaksud dengan gen, individu, populasi, generasi dalam algoritma evolusi!**

**Jawab:**

- Gen: Gen adalah unit informasi terkecil dalam kromosom yang mewakili variabel atau karakteristik dari solusi. Dalam algoritma evolusi, gen diwakili oleh nilai-nilai tertentu seperti biner (0 atau 1), integer, atau bilangan pecahan (real) yang menyusun solusi.
- Individu: Individu adalah kromosom lengkap yang terdiri dari serangkaian gen dan mewakili sebuah solusi potensial dalam ruang pencarian. Dalam konteks algoritma evolusi, individu adalah satu kandidat solusi yang akan dievaluasi kualitasnya.
- Populasi: Populasi adalah kumpulan individu yang ada pada suatu generasi. Populasi digunakan sebagai himpunan solusi yang akan diproses dalam siklus evolusi untuk menghasilkan solusi baru. Ukuran populasi menentukan berapa banyak solusi yang dievaluasi pada tiap generasi.
- Generasi: Generasi adalah satu siklus lengkap dari proses evolusi, mulai dari inisialisasi, reproduksi, seleksi, hingga pembentukan populasi baru. Generasi menggambarkan satu tahap iterasi dalam proses pencarian solusi. Pada setiap

generasi, individu-individu yang lebih baik dipertahankan dan diteruskan ke generasi berikutnya

**3. Apa yang dimaksud dengan fungsi fitness?**

**Jawab:**

Fungsi fitness adalah suatu fungsi yang digunakan untuk mengukur kualitas individu dalam populasi berdasarkan seberapa baik solusi yang direpresentasikannya dalam memenuhi tujuan optimasi. Fitness digunakan untuk mengevaluasi individu dan menentukan peluangnya untuk dipilih dalam proses seleksi. Contohnya dalam masalah maksimasi, nilai fitness bisa diambil dari nilai fungsi objektif yang hendak dimaksimalkan. Semakin tinggi nilai fitness, semakin baik individu tersebut sebagai solusi potensial.

**4. Apa yang dimaksud dengan pernyataan bahwa algoritma evolusi bersifat stochastic?**

**Jawab:**

Algoritma evolusi bersifat stochastic karena algoritma ini menggunakan elemen-elemen acak dalam proses pencarian solusinya, seperti pemilihan individu untuk seleksi, titik potong crossover, dan titik mutasi yang semuanya melibatkan bilangan acak. Konsekuensi dari sifat stochastic ini adalah hasil akhir dari algoritma bisa berbeda-beda setiap kali dijalankan, meskipun permasalahannya sama. Hal ini membuat algoritma evolusi mampu mengeksplorasi ruang solusi yang lebih luas dan berpeluang lebih besar untuk menemukan solusi optimal global dibandingkan algoritma deterministik yang hanya mengikuti jalur pencarian yang tetap

**5. Jelaskan perbedaan antara Soft Constraints dan Hard Constraints, dan berikan contohnya!**

**Jawab:**

**a. Hard Constraints**

Hard constraints adalah kendala-kendala yang bersifat mutlak dan harus dipenuhi. Jika hard constraint dilanggar, maka solusi yang dihasilkan menjadi tidak valid dan tidak dapat digunakan. Hard constraint seringkali digunakan untuk membatasi solusi dalam memenuhi kondisi yang penting dan tidak dapat dinegosiasikan.

Contoh Hard Constraints: Pada masalah penjadwalan kuliah, seorang dosen tidak boleh dijadwalkan mengajar lebih dari satu kelas pada waktu yang sama. Jika aturan ini dilanggar, maka jadwal tersebut tidak valid.

**b. Soft Constraints**

Soft constraints adalah kendala-kendala yang idealnya dipenuhi, tetapi jika dilanggar tidak menyebabkan solusi menjadi tidak valid, hanya mengurangi kualitasnya. Pelanggaran terhadap soft constraint akan menghasilkan solusi yang mungkin masih dapat diterima, tetapi kualitasnya kurang baik.

Contoh Soft Constraints: Pada penjadwalan kuliah, idealnya dosen tidak dijadwalkan mengajar lebih dari dua kelas berturut-turut untuk menghindari kelelahan. Namun, jika dosen tersebut dijadwalkan mengajar tiga kelas berturut-turut, solusinya tetap valid, hanya kualitas jadwal yang dihasilkan kurang baik

## TUGAS KELOMPOK

### 1. Sebutkan semua proses utama dalam siklus algoritma genetika?

Jawab:

#### a. Inisialisasi (Initialization)

- Proses inisialisasi diawali dengan pembuatan himpunan solusi awal yang disebut populasi, di mana setiap individu dalam populasi tersebut direpresentasikan dalam bentuk kromosom. Kromosom ini dapat berupa string biner, angka pecahan (real number), atau format lain, tergantung bagaimana solusi didefinisikan untuk permasalahan yang sedang dihadapi.
- Pada tahap ini, ukuran populasi (popSize) dan panjang kromosom (stringLen) harus ditentukan. Panjang kromosom biasanya bergantung pada tingkat presisi variabel yang digunakan dalam solusi.
- Individu-individu awal dibangkitkan secara acak untuk memberikan keragaman pada populasi awal, yang nantinya akan mengalami proses evolusi.

#### b. Reproduksi (Reproduction)

Tahap reproduksi dilakukan untuk menghasilkan individu baru (offspring) yang merupakan hasil kombinasi dari dua individu "orang tua" (parents) terpilih dari populasi. Proses ini bertujuan untuk menggabungkan karakteristik dari kedua orang tua agar keturunan yang dihasilkan memiliki kualitas yang lebih baik. Reproduksi mencakup dua operator utama, yaitu Crossover dan Mutasi:

- Crossover (Tukar Silang)  
Crossover dilakukan dengan memilih dua individu dari populasi, kemudian menukar sebagian gen dari kedua individu tersebut pada titik potong tertentu (*cut point*) untuk menghasilkan dua individu baru. Proses ini disebut sebagai *one-point crossover*. Selain itu, ada metode crossover lain seperti *two-point crossover* dan *uniform crossover*.
- Mutasi (Mutation)  
Mutasi dilakukan dengan mengubah nilai gen pada satu titik acak di dalam kromosom, untuk mempertahankan keragaman genetik pada populasi. Proses ini berfungsi untuk menghindari keturunan yang seragam, yang dapat menyebabkan algoritma terjebak pada solusi lokal.

#### c. Evaluasi (Evaluation)

- Setiap individu yang telah dihasilkan dari proses crossover dan mutasi harus dievaluasi menggunakan fungsi fitness untuk menilai seberapa baik solusi yang diwakili oleh individu tersebut.
- Fungsi fitness memberikan nilai yang menunjukkan kualitas setiap individu dalam populasi. Semakin tinggi nilai fitness, semakin baik individu tersebut.

#### d. Seleksi (Selection)

Seleksi dilakukan untuk memilih individu yang akan bertahan hidup dan menjadi orang tua pada generasi berikutnya. Proses seleksi ini sangat penting karena individu dengan nilai fitness yang lebih tinggi akan memiliki peluang lebih besar untuk diturunkan ke generasi berikutnya. Beberapa metode seleksi yang umum digunakan:

- **Roulette Wheel Selection**

Pada metode ini, probabilitas terpilihnya individu sebanding dengan nilai fitness yang dimilikinya. Proses ini menyerupai putaran roda roulette, di mana setiap individu mendapatkan bagian sesuai dengan proporsi nilai fitness.

- **Roulette Wheel Selection**

Pada seleksi ini, beberapa individu dipilih secara acak dari populasi, kemudian individu dengan nilai fitness terbaik dari kelompok tersebut dipilih sebagai orang tua.

- **Elitism**

Elitism adalah metode di mana individu dengan nilai fitness terbaik dalam generasi saat ini dipastikan untuk dibawa ke generasi berikutnya, guna mempertahankan kualitas solusi.

## 2. Jelaskan keunggulan algoritma genetika sebagai algoritma yang berbasis populasi!

### Jawab:

Sebagai algoritma yang berbasis populasi, algoritma genetika memiliki keunggulan sebagai berikut.

- **Mampu Menangani Ruang Pencarian yang Luas dan Kompleks**

Algoritma genetika dapat digunakan untuk masalah optimasi dengan ruang pencarian (search space) yang sangat luas dan kompleks. Hal ini karena algoritma bekerja dengan banyak individu sekaligus dalam satu populasi, memungkinkan pencarian solusi dilakukan di berbagai titik secara paralel. Dengan begitu, algoritma ini lebih efisien dalam menjelajahi ruang solusi yang besar dibandingkan metode pencarian tunggal yang hanya berfokus pada satu solusi dalam satu waktu.

- **Menghindari Jebakan Optimum Lokal**

Algoritma genetika memiliki kemampuan untuk melompat keluar dari daerah optimum lokal. Ini berkat adanya mekanisme seperti mutasi yang berfungsi memperkenalkan variasi acak dalam populasi. Mutasi membuat algoritma genetika tetap dapat mengeksplorasi bagian lain dari ruang solusi yang belum terjangkau, menghindari terjebaknya algoritma pada solusi suboptimal yang sering terjadi pada metode deterministik lainnya.

- **Kemampuan Eksploitasi dan Eksplorasi Ruang Solusi yang Efektif**

Algoritma genetika dapat menjaga keseimbangan antara eksplorasi dan eksploitasi. Eksplorasi dilakukan melalui proses mutasi yang memperkenalkan gen baru secara acak, sedangkan eksploitasi dilakukan melalui crossover yang menggabungkan karakteristik baik dari individu-individu yang terpilih. Hal ini memungkinkan

algoritma menemukan solusi optimal dengan lebih efektif, karena tidak hanya fokus pada pengembangan solusi yang sudah ada, tetapi juga menciptakan solusi baru yang potensial.

- **Ketahanan terhadap Kondisi Permasalahan yang Kompleks**

Algoritma genetika dapat diterapkan pada berbagai jenis permasalahan, baik yang bersifat diskrit maupun kontinu, serta mampu menangani fungsi objektif yang tidak beraturan, tidak mulus, atau bahkan yang tidak memiliki turunan (derivatif). Ini karena algoritma genetika tidak memerlukan informasi tambahan mengenai karakteristik fungsi seperti metode optimasi berbasis gradien, sehingga lebih fleksibel digunakan untuk berbagai jenis masalah.

- **Kemampuan Menemukan Beberapa Solusi Optimal dalam Fungsi Multimodal**

Algoritma genetika mampu menemukan beberapa solusi optimal secara bersamaan pada permasalahan dengan fungsi multimodal, yaitu fungsi yang memiliki lebih dari satu titik optimal. Karena algoritma bekerja dengan banyak individu sekaligus dalam populasi, beberapa solusi dapat ditemukan pada generasi yang sama tanpa harus mengeliminasi solusi-solusi yang berbeda namun optimal.

- **Paralelisme dan Pemrosesan Sub-Populasi**

Algoritma genetika dapat dengan mudah diparalelisasi karena evaluasi setiap individu dalam populasi dapat dilakukan secara independen satu sama lain. Hal ini memungkinkan setiap sub-populasi diproses pada sejumlah komputer secara paralel, yang akan mengurangi waktu komputasi dan meningkatkan efisiensi pencarian. Sub-populasi ini juga dapat diterapkan pada satu komputer untuk menjaga keragaman populasi dan meningkatkan kualitas solusi yang dihasilkan.

- **Kestabilan dan Robustness dalam Menangani Ruang Solusi yang Besar**

Algoritma genetika termasuk metode yang robust karena dapat menghadapi ruang solusi yang besar dan kompleks tanpa kehilangan stabilitas. Algoritma ini cenderung lebih tahan terhadap perubahan kecil pada data atau parameter, sehingga mampu mempertahankan kualitas pencarian solusi yang baik meskipun ruang solusi sangat bervariasi.

Dengan berbagai keunggulan tersebut, algoritma genetika menjadi salah satu metode optimasi berbasis populasi yang sangat powerful dan fleksibel untuk berbagai jenis permasalahan, terutama yang memiliki ruang pencarian besar, fungsi objektif yang rumit, serta kebutuhan untuk menemukan beberapa solusi optimal sekaligus.

### 3. Jelaskan apa yang dimaksud dengan algoritma genetika bersifat ergodic?

**Jawab:**

Algoritma genetika dikatakan bersifat *ergodic* karena algoritma ini memiliki kemampuan untuk mengeksplorasi seluruh ruang solusi dalam jangka waktu yang panjang. Dalam konteks teori sistem dinamis, ergodic berarti bahwa algoritma tersebut, jika dibiarkan berjalan cukup lama dan dengan variasi parameter yang tepat, dapat mengunjungi atau

menjelajahi setiap titik solusi yang mungkin dalam ruang pencarian, sehingga memberikan peluang untuk menemukan solusi optimal global. Kemampuan ini didukung oleh proses crossover dan mutasi yang memperkenalkan variasi pada gen individu dalam populasi, memungkinkan algoritma menjelajahi bagian-bagian ruang solusi yang sebelumnya belum terjangkau. Dengan demikian, sifat ergodic membantu algoritma genetika untuk menjaga keberagaman populasi dan memastikan bahwa algoritma tidak hanya berkutat di sekitar solusi lokal, tetapi juga terus menjelajahi area lain untuk mencapai solusi optimal. Karena alasan ini, algoritma genetika lebih fleksibel dan efektif dalam menghindari jebakan pada solusi suboptimal, membuatnya lebih unggul dalam menemukan solusi optimal pada ruang pencarian yang luas dan kompleks dibandingkan metode konvensional yang sering terjebak pada titik optimum lokal.

**4. Iterasi GAs diulang terus sampai kondisi berhenti tercapai. Jelaskan beberapa kriteria untuk hal ini?**

**Jawab:**

Iterasi pada algoritma genetik akan diulang terus hingga kondisi berhenti tercapai. Kondisi berhenti dapat dipenuhi melalui beberapa kriteria, yaitu:

- **Jumlah Generasi Tertentu Telah Terpenuhi (*Fixed Generations*)**

Algoritma akan melakukan iterasi dan kemudian berhenti pada generasi  $n$ , dimana nilai  $n$  ditentukan sebelumnya berdasarkan beberapa eksperimen pendahuluan. Semakin tinggi ukuran dan kompleksitas masalah, maka nilai  $n$  akan semakin besar. Nilai  $n$  ditentukan sedemikian rupa sehingga konvergensi pada populasi dapat tercapai dan akan sulit didapatkan solusi yang lebih baik setelah  $n$  iterasi.

- **Nilai Fitness Maksimum Telah Tercapai (*Fitness Threshold*)**

Iterasi akan berhenti dilakukan ketika nilai *fitness* maksimum telah tercapai. Hal ini memiliki arti bahwa nilai *fitness* dari individu terbaik di populasi telah mencapai atau melebihi ambang batas (*threshold*) yang ditentukan. Nilai *fitness* maksimum menunjukkan bahwa solusi yang baik telah ditemukan.

- **Kondisi Stagnasi (*Convergence*)**

Jika tidak terdapat perubahan yang signifikan dalam nilai *fitness* populasi selama beberapa generasi, iterasi dapat berhenti dilakukan. Hal ini menunjukkan bahwa algoritma genetik kesulitan untuk mendapatkan solusi yang lebih baik, sehingga penambahan iterasi hanya akan membuang waktu. Solusi yang cukup baik telah ditemukan.

- **Waktu Maksimum (*Time Limit*) Telah Tercapai**

Iterasi algoritma akan berhenti setelah  $t$  satuan tercapai, yaitu ketika iterasi telah mencapai batas waktu yang ditentukan. Hal ini biasa digunakan untuk keperluan perbandingan performa dari beberapa algoritma. Selain itu, hal ini umum dilakukan ketika terdapat kendala pada waktu komputasi.



- **Evaluasi Individu Maksimum (*Maximum Evaluations*) Tercapai**

Algoritma dapat berhenti melakukan iterasi ketika telah melakukan sejumlah evaluasi *fitness* tertentu pada setiap individu, terlepas dari jumlah generasi yang telah tercapai selama algoritma dijalankan.

- **Tingkat Mutasi atau Keragaman Rendah (*Diversity Threshold*)**

Jika keragaman dalam populasi yang dihasilkan terlalu rendah, misalnya pada kasus dimana semua individu menjadi sangat mirip, algoritma dapat berhenti melakukan iterasi. Hal ini dikarenakan kemungkinan untuk solusi yang lebih baik dapat ditemukan menurun drastis yang kemudian menuju pada kondisi stagnasi atau konvergen.

**5. Misalkan P1 dan P2 adalah parent untuk proses crossover. Tentukan offspring yang terbentuk jika dilakukan one-cut-point crossover pada titik ke-5.**

**P1 [ 0 0 1 1 0 0 1 ]**

**P2 [ 1 0 0 1 1 1 1 ]**

**Jawab:**

Pada proses *one-cut-point crossover*, kromosom dari kedua *parent* akan dipotong pada titik yang ditentukan, dalam hal ini titik ke-5, dan bagian setelah titik ke-5 akan dipertukarkan.

**1. Menentukan nilai pc**

Pada soal ini, kami tentukan nilai pc yang akan digunakan yaitu  $pc = 0.8$ . Hal ini memiliki arti bahwa kemungkinan *crossover* terjadi sebanyak 80%. Kemudian, dapat dihitung rasio *offspring* yang akan dihasilkan yaitu  $0.8 \times 2 = 1.6$ . Ketika dibulatkan ke bawah, dapat diketahui bahwa akan dihasilkan sebanyak 1 *offspring* dari proses *crossover*. Namun, setiap proses *crossover* pada satu pasang *parent* akan selalu menghasilkan 2 *offspring*. Karena *crossover* pasti terjadi pada perhitungan ini, maka *offspring* yang akan dihasilkan adalah 2.

**2. Memotong Kromosom di Titik ke-5**

- $P_1$  : Bagian sebelum titik ke-5 adalah [00110], dan bagian setelahnya adalah [01].
- $P_2$  : Bagian sebelum titik ke-5 adalah [10011], dan bagian setelahnya adalah [11].

**3. Menukar Bagian Setelah Titik ke-5**

- *Offspring 1*

*Offspring 1* mengambil bagian awal dari  $P_1$  dan bagian akhir dari  $P_2$ , sehingga didapatkan hasil:

$$\text{Offspring 1} = [00110] + [11] = [0011011]$$

- *Offspring 2*

*Offspring 2* mengambil bagian awal dari  $P_2$  dan bagian akhir dari  $P_1$ , sehingga didapatkan hasil:

$$\text{Offspring 2} = [10011] + [01] = [1001101]$$

Setelah dilakukan proses *one-cut-point crossover* pada titik ke-5 pada  $P_1$  dan  $P_2$ , dihasilkan dua *offspring* yaitu **[0011011]** dan **[1001101]**

6. Misalkan P adalah parent untuk proses mutasi. Tentukan offspring yang terbentuk jika dilakukan mutasi pada titik ke-5.

**P      [ 0 0 1 1 0 0 1 ]**

**Jawab:**

Dalam proses mutasi pada algoritma genetika, mutasi berarti membalik bit pada posisi yang ditentukan. Jika bit pada posisi tersebut adalah 0, maka akan berubah menjadi 1, dan sebaliknya, jika bit pada posisi tersebut adalah 1, maka akan berubah menjadi 0.

Posisi ke-5 dari string P adalah 0 (dimulai dari kiri dengan posisi pertama dihitung sebagai 1). Ketika bit ini dibalik dari 0 menjadi 1, string yang dihasilkan (offspring) akan menjadi:

$$\text{Offspring (C)} = [0011101]$$

Jadi, setelah mutasi pada posisi ke-5, bit tersebut berubah dari 0 menjadi 1, menghasilkan offspring yang baru yakni **C = [0011101]**.

7. Untuk masalah maksimasi (mencari nilai maksimum) dari sebuah fungsi sebagai berikut  $\max, y = f(x) = -(x^2)/2 + 4x + 40, 0 \leq x \leq 15$  Lengkapi tabel berikut:

	chromosome	x	$y=f(x)$	fitness
$P_1$	[ 0 0 0 1 ]			
$P_2$	[ 1 1 0 0 ]			
$P_3$	[ 1 0 1 1 ]			
$P_4$	[ 1 1 1 1 ]			

**Jawab:**

1. Konversi Kromosom ke Nilai x

- $P_1 = [0001] \rightarrow x = 1$
- $P_2 = [1100] \rightarrow x = 12$
- $P_3 = [1011] \rightarrow x = 11$
- $P_4 = [1111] \rightarrow x = 15$

## 2. Substitusi ke Fungsi $f(x)$

Substitusi nilai  $x$  ke dalam persamaan  $y = f(x) = (-\frac{x^2}{2}) + 4x + 40$ :

- Untuk  $x = 1$ :

$$f(1) = (-\frac{1^2}{2}) + 4(1) + 40 = -\frac{1}{2} + 4 + 40 = 43.5$$

- Untuk  $x = 12$ :

$$f(12) = (-\frac{12^2}{2}) + 4(12) + 40 = -72 + 48 + 40 = 16$$

- Untuk  $x = 11$ :

$$f(11) = (-\frac{11^2}{2}) + 4(11) + 40 = -60.5 + 44 + 40 = 23.5$$

- Untuk  $x = 15$ :

$$f(15) = (-\frac{15^2}{2}) + 4(15) + 40 = -112.5 + 60 + 40 = -12.5$$

## 3. Tabel Lengkap

chromosome	$x$	$y = f(x)$	fitness
$P_1 = [0001]$	1	43.5	43.5
$P_2 = [1100]$	12	16	16
$P_3 = [1011]$	11	23.5	23.5
$P_4 = [1111]$	15	-12.5	-12.5