

Pertemuan 11

Algoritma Genetika

A. Konsep Evolusi dan Seleksi Alam

Algoritma Genetika (Genetic Algorithm, GA) terinspirasi dari teori evolusi biologis yang dikemukakan oleh Charles Darwin, khususnya prinsip bahwa organisme yang paling sesuai (fit) dengan lingkungannya akan memiliki peluang lebih besar untuk bertahan hidup dan berkembang biak. Dalam konteks komputasi, konsep ini diterjemahkan menjadi proses pencarian solusi optimal melalui mekanisme evolusi buatan.

1. Evolusi sebagai Proses Optimasi

Dalam biologi, evolusi terjadi melalui perubahan bertahap dalam populasi makhluk hidup. Individu yang memiliki sifat lebih baik akan lebih mungkin bertahan dan menghasilkan keturunan. Dalam Algoritma Genetika, solusi masalah diwakili sebagai individu dalam suatu populasi, dan GA menghasilkan solusi yang lebih baik melalui proses iteratif yang menyerupai evolusi: seleksi → persilangan → mutasi.

Ciri utama evolusi dalam GA:

- Terdapat populasi yang berisi banyak kandidat solusi.
- Setiap individu memiliki kromosom (representasi karakteristik solusi).
- Individu mengalami proses “reproduksi” untuk menghasilkan keturunan.
- Populasi secara bertahap berkembang menuju solusi yang lebih optimal.

2. Prinsip Seleksi Alam

Konsep inti seleksi alam adalah:

"Yang paling fit akan memiliki peluang terbesar untuk bertahan dan menghasilkan keturunan."



Dalam GA, tingkat *fitness* menentukan kualitas sebuah solusi. Individu dengan nilai *fitness* tinggi:

- lebih mungkin dipilih untuk reproduksi,
- memiliki probabilitas lebih besar untuk mewariskan karakteristiknya,
- mempercepat konvergensi menuju solusi optimal.

Seleksi alam dalam GA dilakukan menggunakan berbagai metode seperti:

- Roulette Wheel Selection — peluang dipilih sebanding dengan *fitness*.
- Tournament Selection — memilih terbaik dari kelompok acak kecil.
- Rank Selection — individu diberi ranking untuk menghindari dominasi nilai *fitness* ekstrem.

3. Analogi Evolusi dan GA

Biologi	Algoritma Genetika
Organisme	Solusi / individu
Gen	Nilai parameter
Kromosom	Representasi solusi
Populasi	Kumpulan solusi
Fitness	Nilai kualitas solusi
Seleksi alam	Pemilihan solusi terbaik
Perkawinan (crossover)	Pertukaran bagian solusi
Mutasi	Perubahan acak kecil pada solusi
Evolusi generasi	Iterasi perbaikan

GA bekerja dengan cara mengombinasikan sifat-sifat baik dari beberapa solusi untuk membentuk solusi baru yang lebih baik, layaknya evolusi biologis.

4. Mengapa Seleksi Alam Efektif dalam AI dan Optimasi?

Algoritma Genetika unggul karena:

- Mampu mencari solusi global, bukan hanya lokal.
- Tidak memerlukan turunan matematis atau bentuk fungsi tertentu.
- Efektif untuk masalah kompleks, seperti penjadwalan, optimasi rute, penentuan parameter model ML, dan desain sistem.



Seleksi alam memastikan bahwa:

- Solusi buruk secara bertahap dieliminasi.
- Solusi baik diwariskan.
- Algoritma menemukan pola optimal secara bertahap melalui generasi.

B. Representasi Kromosom dan Fungsi Fitness

1. Representasi Kromosom

Dalam Algoritma Genetika (Genetic Algorithm/GA), *kromosom* adalah bentuk representasi solusi dari suatu masalah. Setiap kromosom terdiri dari beberapa *gen*, yaitu elemen yang menyimpan nilai variabel dari solusi tersebut. Cara merepresentasikan kromosom sangat bergantung pada jenis masalah yang ingin diselesaikan.

Bentuk Representasi Kromosom yang Umum

1. Representasi Biner (*Binary Encoding*)

- Gen berupa angka 0 dan 1.
- Cocok untuk optimasi sederhana, pemilihan fitur, atau masalah logika.
- Contoh:

Kromosom: 101011

Interpretasi: setiap bit menunjukkan status fitur/variabel (aktif atau tidak).

2. Representasi Bilangan Real (Real-Valued Encoding)

- Gen berupa angka pecahan atau bilangan real.
- Cocok untuk optimasi fungsi matematis atau parameter algoritma.
- Contoh:

[0.25, 3.41, 1.87] → mewakili tiga parameter dalam model.

3. Representasi Permutasi (Permutation Encoding)

- Digunakan untuk masalah yang berbasis urutan (ordering problem).
- Sangat populer untuk:
 - Travelling Salesman Problem (TSP)
 - Penjadwalan



- Contoh:
[3, 1, 4, 2, 5] → urutan kota/pekerjaan yang harus dikunjungi/diselesaikan.
- 4. Representasi Simbolik atau String Karakter
 - Digunakan untuk sistem yang lebih kompleks seperti ekspresi matematika, program (GP), atau pola tertentu.
 - Contoh:
"ABCDAB"

Struktur Kromosom

Kromosom pada GA umumnya bersifat:

- Tetap panjangnya (fixed length).
- Mewakili satu solusi lengkap untuk masalah yang dianalisis.

2. Fungsi Fitness (Fitness Function)

Fungsi fitness adalah inti utama dari Algoritma Genetika, karena menentukan kualitas suatu solusi. Fitness adalah nilai kuantitatif yang menggambarkan seberapa “baik” suatu individu dalam menyelesaikan masalah.

Tujuan Fungsi Fitness

- Mengukur kualitas solusi.
- Menentukan peluang individu untuk dipilih dalam **seleksi alam**.
- Mengarahkan proses evolusi menuju solusi optimal.

3. Karakteristik Fungsi Fitness yang Baik

Fungsi fitness harus:

- Merefleksikan tujuan optimasi, misalnya meminimalkan biaya, memaksimalkan keuntungan.
- Menghasilkan nilai numerik yang dapat dibandingkan antar individu.
- Mampu membedakan (discriminative) solusi yang baik dan yang buruk.
- Stabil, tidak terlalu sensitif terhadap perubahan kecil.



4. Contoh Fungsi Fitness untuk Berbagai Masalah

A. Optimasi Fungsi Matematis

Misal ingin *memaksimalkan* fungsi berikut:

$$f(x) = x^2$$

Maka fitness = (x^2) .

B. Travelling Salesman Problem (TSP)

Tujuan: *meminimalkan* total jarak tempuh.

Namun GA cenderung menggunakan **fitness yang harus dimaksimalkan**, sehingga sering diubah menjadi:

$$\text{Fitness} = \frac{1}{\text{Total Jarak}}$$

Semakin kecil jarak → semakin besar nilai fitness → semakin baik.

C. Penjadwalan

Fitness dapat berupa:

- Minimasi total waktu pengerjaan (makespan)
- Minimasi konflik jadwal
- Minimasi keterlambatan

Contoh:

$$\text{Fitness} = \frac{1}{\text{Jumlah Konflik} + 1}$$

D. Seleksi Fitur pada Machine Learning

Misal memilih kombinasi fitur terbaik untuk klasifikasi:

$$\text{Fitness} = \text{Accuracy Model K-NN}$$

Semakin tinggi akurasi → semakin besar fitness.

5. Hubungan Kromosom dan Fitness dalam Evolusi GA

Setiap generasi GA mengikuti mekanisme berikut:

1. Kromosom dibangkitkan (populasi awal).

Kecerdasan Buatan (Artificial Intelligence)



2. Fitness dihitung untuk setiap individu.
3. Individu terbaik dipilih untuk proses reproduksi.
4. Proses crossover dan mutasi menciptakan solusi baru.
5. Populasi berkembang menuju solusi semakin optimal.

Fitness berperan seperti *nilai adaptasi* dalam seleksi alam:

- Individu dengan fitness tinggi → lebih banyak disalin/memiliki keturunan.
- Individu dengan fitness rendah → berkurang atau tereleminasi.

C. Operator GA: Seleksi, Crossover, Mutasi

Operator dalam Algoritma Genetika adalah mekanisme kunci yang memungkinkan terjadinya **evolusi**, yaitu proses menghasilkan solusi baru yang lebih baik melalui seleksi, persilangan (crossover), dan mutasi. Operator-operator ini meniru mekanisme biologis pada organisme hidup.

1. Seleksi (Selection)

Seleksi adalah proses memilih individu yang akan menjadi *orang tua (parents)* dalam menghasilkan generasi berikutnya. Individu dengan fitness lebih tinggi mendapatkan peluang lebih besar untuk terpilih—mewakili konsep seleksi alam Darwin: *survival of the fittest*.

Tujuan Seleksi

- Memastikan individu terbaik berpeluang lebih besar melanjutkan keturunannya.
- Mengatur tekanan evolusi agar pencarian solusi lebih cepat dan tidak stagnan.

Metode Seleksi yang Umum Digunakan

a. Roulette Wheel Selection

- Peluang terpilih proporsional dengan nilai fitness.
- Setiap individu seperti “bagian roda”, semakin besar fitness → semakin besar bagian roda.



- Cocok untuk masalah dengan variasi fitness yang moderat.

b. Tournament Selection

- Beberapa individu dipilih secara acak.
- Individu terbaik dalam kelompok kecil tersebut yang menjadi pemenang.
- Keunggulan: sederhana dan dapat mengontrol tekanan seleksi melalui ukuran turnamen.

c. Rank Selection

- Individu diurutkan berdasarkan ranking fitness, bukan nilai fitness absolut.
- Menghindari dominasi individu dengan fitness terlalu tinggi.

2. Crossover (Rekombinasi)

Crossover merupakan operator biologis yang meniru proses *rekombinasi genetik* ketika reproduksi. Dua kromosom induk bertukar bagian tertentu untuk menghasilkan dua individu baru (offspring).

Tujuan Crossover

- Menggabungkan sifat-sifat terbaik dari dua individu.
- Mengeksplorasi ruang solusi baru.
- Meningkatkan keberagaman populasi.

Jenis-Jenis Crossover

a. Single-Point Crossover

- Dua induk dipotong pada satu titik.
- Segmen setelah titik tersebut ditukar.

Contoh:

Parent1: 110|101

Parent2: 011|001

Offspring: 110001, 011101



b. Two-Point Crossover

- Dua titik potong.
- Bagian di antara dua titik dipertukarkan.

c. Uniform Crossover

- Setiap gen memiliki peluang tertentu (mis. 0.5) untuk ditukar.
- Menghasilkan kombinasi yang lebih variatif.

d. Order Crossover (OX) — untuk TSP atau permutasi

Digunakan pada representasi permutasi (urutan), misalnya optimasi rute.

Menjaga agar offspring tidak memiliki elemen duplikat.

3. Mutasi (Mutation)

Mutasi adalah operator yang mengubah satu atau lebih gen dalam kromosom secara acak. Tujuan utama mutasi adalah **menambah variasi genetik** agar populasi tidak terjebak pada solusi lokal (local optimum).

Tujuan Mutasi

- Menjaga keragaman populasi.
- Mencegah stagnasi evolusi.
- Memungkinkan eksplorasi solusi baru yang tidak mungkin tercapai hanya melalui crossover.

Jenis Mutasi yang Umum

a. Bit Flip Mutation (untuk representasi biner)

Gen '0' menjadi '1' atau sebaliknya.

Contoh: 101011 → 100011



b. Swap Mutation (untuk permutasi)

Dua posisi dalam kromosom ditukar.

Cocok untuk masalah penjadwalan dan TSP.

Contoh: [1, 3, 5, 2] → [1, 2, 5, 3]

c. Random Resetting (untuk bilangan real)

Satu gen diubah menjadi nilai acak baru dalam rentang tertentu.

d. Scramble dan Inversion Mutation

Digunakan saat kromosom berupa urutan:

- **Scramble:** bagian tertentu diacak.
- **Inversion:** urutan dibalik.

4. Peran Seleksi, Crossover, dan Mutasi dalam Evolusi GA

Operator	Fungsi Utama	Dampak pada Evolusi
Seleksi	Memilih individu terbaik	Meningkatkan kualitas populasi
Crossover	Menggabungkan sifat terbaik	Eksplorasi solusi baru dan perbaikan kualitas
Mutasi	Menambah keragaman	Mencegah konvergensi prematur

Ketiga operator ini bekerja secara sinergis membentuk siklus evolusi GA:

1. Populasi awal dibentuk.
2. Fitness dihitung.
3. Seleksi orang tua dilakukan.
4. Crossover menghasilkan individu baru.
5. Mutasi memberikan variasi tambahan.
6. Dibentuk populasi baru untuk generasi berikutnya.
7. Proses berlanjut hingga tercapai solusi optimal.

