# Contoh Kasus Algoritma Genetika: Optimasi Rute Pengiriman Barang

## 1. Kasus yang lebih spesifik

Optimasi rute pengiriman barang untuk perusahaan logistik skala menengah yang memiliki 1 depot dan beberapa kendaraan. Tujuan: meminimalkan total jarak/tempuh dan jumlah kendaraan yang digunakan sambil memenuhi kapasitas kendaraan dan batas waktu pengiriman.

## 2. Representasi kasus dalam kombinasi gen-gen

Kromosom direpresentasikan sebagai urutan (permutasi) daftar pelanggan yang harus dikunjungi, dipisahkan menurut kendaraan. Setiap gen merepresentasikan satu pelanggan (customer ID).   
  
Contoh kromosom (dua kendaraan):  
[ Depot | C5, C2, C8 | Depot ; Depot | C1, C3, C4, C6, C7 | Depot ]  
  
Alternatif representasi: matriks biner (customer × vehicle) atau encoding daftar rute variabel.

## 3. Isi gen

- Customer ID (mis. C1..C10)  
- Koordinat lokasi (latitude, longitude) sebagai atribut eksternal untuk menghitung jarak  
- Demand (volume/berat) untuk masing-masing customer  
- Waktu jendela pengiriman (opsional): [earliest, latest]  
  
Contoh daftar pelanggan:  
{C1, C2, C3, C4, C5, C6, C7, C8, C9, C10}

## 4. Berapa kombinasinya (jelasnya dari mana jawaban anda)

Untuk masalah rute dengan N pelanggan dan asumsi satu kendaraan (permutasi penuh):  
Jumlah kemungkinan = N! (faktorial N).   
  
Contoh numerik:  
- N = 8 → 8! = 40.320  
- N = 10 → 10! = 3.628.800  
- N = 12 → 12! = 479.001.600  
  
Jika ada K kendaraan, ruang solusi dibagi ke rute-rute sehingga jumlah kombinasi jauh lebih besar (komposisi + permutasi). Oleh karena itu pendekatan brute-force cepat menjadi tidak praktis.

## 5. Apa alasan kasus tersebut tidak dapat diselesaikan secara acak atau berurutan saja?

Ruang solusi tumbuh faktorial terhadap jumlah pelanggan. Brute-force (menguji semua rute) menjadi tak terjangkau untuk N>12. Pencarian acak tanpa panduan kemungkinan besar menemukan solusi buruk atau melanggar constraint (mis. kapasitas). Algoritma Genetika memberikan mekanisme eksplorasi dan eksploitasi yang efisien: seleksi solusi baik, crossover yang mempertahankan sub-rute berguna, dan mutasi untuk diversifikasi.

## 6. Apa persyaratannya (constraints)

- Setiap pelanggan harus dikunjungi tepat satu kali.  
- Kapasitas kendaraan tidak boleh terlampaui (sum demand pada rute ≤ kapasitas).  
- Setiap rute mulai dan berakhir di depot.  
- Jika ada time windows: pelayanan harus berada dalam jendela waktu [earliest, latest]. Pelanggaran dikenai penalti.  
- Batasan durasi kerja supir (maksimum jam per rute).  
- Batasan jumlah kendaraan tersedia (K maksimum).

## 7. Termasuk jenis AG apa?

Jenis: \*\*Rute\*\* (Vehicle Routing Problem — VRP).   
  
Penjelasan: Masalah ini adalah variasi dari Travelling Salesman Problem (TSP) yang diperluas dengan kapasitas kendaraan, time windows, dan banyak kendaraan. GA tipe rute sering menggunakan encoding permutasi dan operator khusus (Order Crossover, Edge Recombination, Insertion Mutation) untuk menjaga validitas rute.

## 8. Keunikan kasus (fokus pada GA)

- \*\*Representasi kromosom\*\*: permutasi pelanggan dengan pemisah rute; menjaga validitas kromosom penting agar tidak ada pelanggan terduplikasi atau terlewat.  
- \*\*Fungsi kecocokan (fitness)\*\*: biasanya kombinasi dari beberapa tujuan dan penalti:  
 • Total jarak/tempuh (di-minimize)  
 • Jumlah kendaraan digunakan (di-minimize)  
 • Penalti untuk melanggar kapasitas atau time windows  
 • Penalti untuk durasi rute yang melebihi batas  
  
Rumus fitness (contoh sederhana, harus disesuaikan):  
fitness = - (w1 \* total\_distance + w2 \* vehicle\_count + w3 \* total\_penalty)  
  
Operator GA khusus:  
- \*\*Crossover\*\*: Order Crossover (OX) atau Edge Recombination untuk mempertahankan urutan pelanggan yang baik.  
- \*\*Mutasi\*\*: Swap mutation, Insertion, atau 2-opt local improvement sebagai mutation-operator hybrid.  
  
Keunikan implementasi: perlu mekanisme repair atau decoding untuk memastikan hasil offspring valid (tidak melanggar kapasitas atau duplikasi). Penalti dan fungsi repair menjadi inti membangun fitness yang efektif.

Dokumen dibuat otomatis pada 2025-10-01 07:23:16