**Menyelesaikan N-Puzzle Menggunakan Genetic Algorithm**

Dibuat untuk Memenuhi Salah Satu Tugas Mata Kuliah Pengantar Sistem Cerdas



Disusun oleh :

Ihsan Fajari 2014730018

Nadhila Larasati 2014730035

Ellena Angelica 2015730029

Evelyn Wijaya 2015730030

Tegar Muhammad 2015730046

**12 November 2017**

**FAKULTAS TEKNOLOGI INFORMASI DAN SAINS**

**UNIVERSITAS KATOLIK PARAHYANGAN**

**BANDUNG**

1. **Definisi Problem**
   1. **State :** (Blank, Puzzle)

N adalah elemen array puzzle Puzzle[ ] dimana setiap i (1, 2, 3, …., N) dalam Puzzle[i] akan merepresentasikan tile number pada posisi i. Posisi yang blank direpresentasikan dengan 0. Posisi dari tile akan diberi nomor dari kiri ke kanan dan atas ke bawah. State Blank mencatat setiap posisi blank.

* 1. **Initial State**

Pada Initial state, setiap tile akan diisikan angka yang acak mulai dari 1 sampai dengan N-1 , posisi (index) angka dan blank juga akan diacak.

* 1. **Actions**

Ada 4 aksi yang dilakukan yaitu moveUp, moveDown, moveLeft, dan moveRight. Aksi moveUp menukarkan tile yang berada tepat di bawah blank tile dengan blank tile. Aksi moveDown menukarkan tile yang berada tepat di atas blank tile dengan blank tile. Aksi moveLeft menukarkan tile yang berada tepat di kiri blank tile dengan blank tile. Aksi moveRight menukaukan tile yang berada tepat di kanan blank tile dengan blank tile.

* 1. **Goal**

Goal state adalah setiap angka pada puzzle telah terurut menaik dari angka paling kecil ke angka paling besar, sebuah puzzle dikatakan telah mencapai goal state apabila blank tile ( tile yang berisi angka 0) tidak berada diantara tile yang bukan merupakan blank tile ( tidak berisi angka 0).

Misal, terdapat inisial state dengan N = 8 seperti dibawah ini :.

|  |  |  |
| --- | --- | --- |
| 2 | 8 | 3 |
| 1 | 6 | 4 |
| 7 | 0 | 5 |

Goal state yang diharapkan adalah seperti dibawah ini.

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 8 | 0 | 4 |
| 7 | 6 | 5 |

Angka 0 tidak berada di lintasan atau (path) antara tile berisi angka 1 sampai tile berisi angka 8.

* 1. **Cost**

Cost atau biaya untuk setiap aksi yang dilakukan adalah 1.

1. **Definisi Genetik**
   1. **Kromosom**

Pada N-Puzzle, kromosom dapat didefinisikan dengan pergeseran tiap petak.Maksudnya, tiap petak yang bergeser akan dicatat menjadi kromosom. Ke arah atas akan direpresentasikan dengan 0, ke bawah adalah 1, ke kiri adalah 2, dan ke kanan adalah 3.

* 1. **Fungsi Fitness**

Pada N-Puzzle, isi kromosom merupakan bagian dari penggerak initial state menuju goal state. Hasil akhir(kondisi inisial state sekarang) dari penggerakan puzzle dengan isi kromosom akan dibandingkan dengan goal state, seberapa banyakah kesamaanya. Ini juga disebut dengan metode missplaced tile.

* 1. **Operator Genetika**

1. Operator Selection

Selection atau Seleksi digunakan untuk memilih individu-individu mana saja yang akan dipilih untuk proses crossover dan mutation. Seleksi digunakan untuk mendapatkan calon induk yang memiliki kualitas yang baik. “Induk yang baik akan menghasilkan keturunan yang baik”. Semakin tinggi nilai fitness suatu individu semakin besar kemungkinannya untuk terpilih.

Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai fitness ini yang nantinya akan digunakan pada tahap-tahap seleksi berikutnya. Masing-masing individu dalam wajah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai obyektif dirinya sendiri terhadap nilai obyektif dari semua individu dalam wadah seleksi tersebut.

2. Operator Crossover

Crossover adalah operator dari algoritma genetika yang melibatkan dua induk untuk menghasilkan kromosom baru. Operasi ini tidak selalu dilakukan pada semua individu yang ada. Prinsip pindah silang ini adalah melakukan operasi (pertukaran,aritmatika) pada gen-gen yang bersesuaian dari dua induk untuk mengasilkan individu baru. Proses crossover dilakuan pada setiap individu dengan probabilitas crossover yang titentukan.

3. Operator Mutation

Operator berikutnya pada algoritma genetika adalah mutasi gen. Operator ini berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Kromosom anak dimutasi dengan menambahkan nilai random yang sangat kecil (ukuran langkah mutasi), dengan probabilitas yang rendah, Peluang mutasi (Pa) didefinisikan sebagai presentasi dari jumlah total gen pada populasi yang mengalami mutasi. Peluang mutasi mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika peluang mutasi terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi bila peluang mutasi ini terlalu besar, maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya, dan juga algoritma kehilangan kemampuan untuk belajar dari histori pencarian.

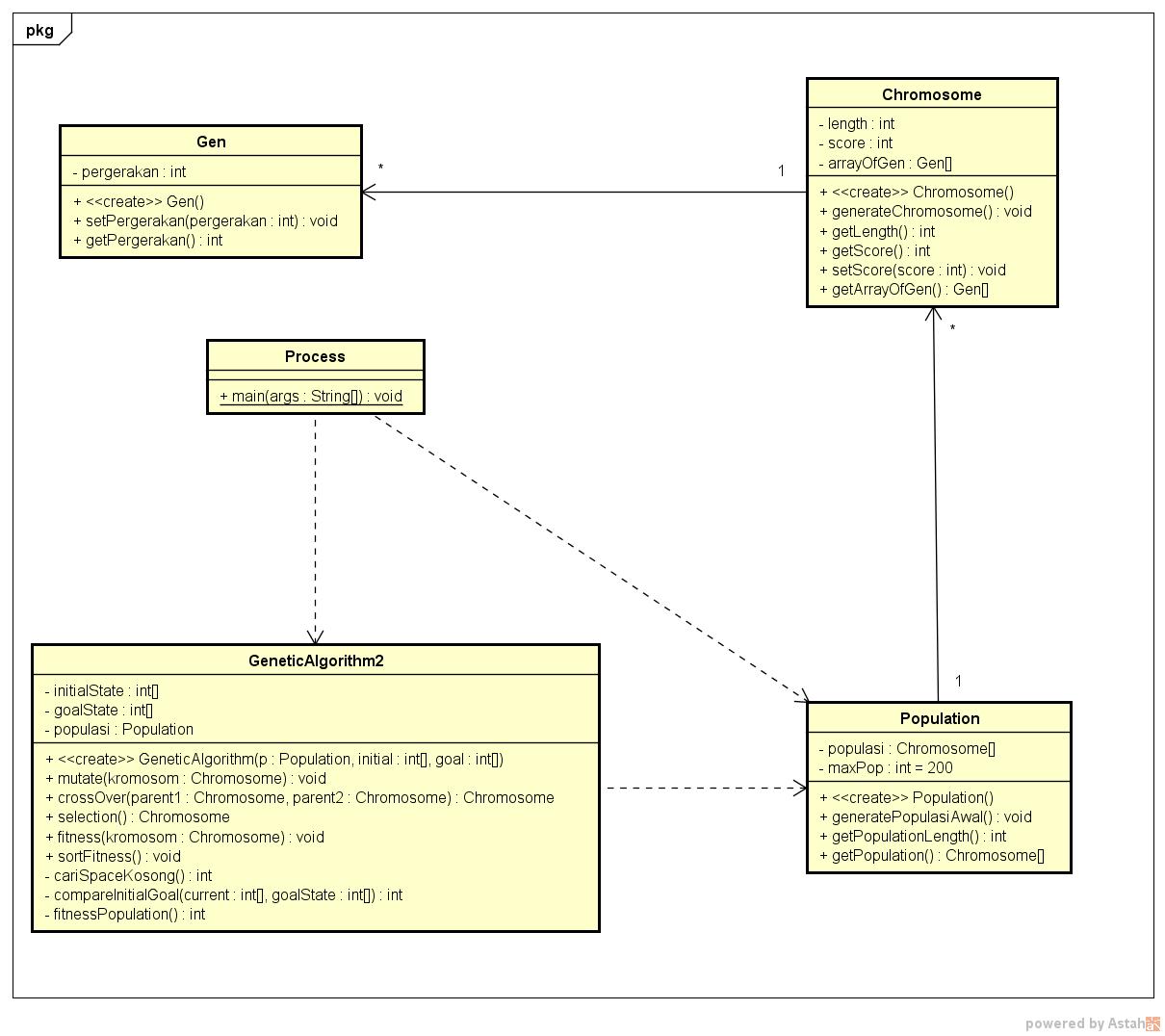
* 1. **Kondisi Berhenti**

N-Puzzle akan berhenti mencari solusi apabila solusi yang ditemukan sudah optimal atau tidak ditemukan solusi untuk test case tersebut sampai waktu tertentu.

* 1. **Varian Genetic Algorithm**

Varian yang digunakan adalah Elitism. Elitism memiliki prinsip "yang terbaik harus bertahan". Individu terbaik dari populasi induk dibandingkan dengan individu terbaik dari populasi keturunan. Yang terbaik di antara kedua populasi ditambahkan ke dalam generasi selanjutnya. Elitisme menjamin fitness generasi berikutnya akan lebih baik atau sama dengan fitness generasi induk.

1. **Diagram Kelas**



1. **Penjelasan Diagram Kelas**

Untuk membuat program N-Puzzle yang menggunakan genetic algorithm, kelompok kami membuat 5 kelas yaitu :

* **Kelas Gen**

Kelas Gen memiliki atribut pergerakan dengan tipe integer. Atribut pergerakan akan meng-generate gen baru dengan cara merandom keluaran angka 0 yang berarti gerak ke atas, 1 berarti gerak ke bawah, 2 berarti gerak ke kiri, dan 3 berarti gerak ke kanan.

* **Kelas Chromosome**

Kelas Chromosome memiliki atribut length dan score yang bertipe integer, dan arrayOfGen bertipe Gen yang disimpan ke dalam array. Kelas ini akan men-generate kromosom pada method generateChromosome().

* **Kelas Population**

Kelas Population memiliki atribut populasi bertipe Chromosome yang disimpan pada array, dan maxPop bertipe integer yang akan diisi maksimal 200 individu. Pada kelas ini, populasi awal akan di-generate pada method generatePopulasiAwal() yang individunya berasal dari Chromosome.

* **Kelas GeneticAlgorithm2**

Kelas GeneticAlgorithm memiliki atribut initialState dan goalState bertipe integer yang disimpan ke dalam array, dan populasi bertipe Population. Kelas ini memiliki method-method yaitu:

* **mutate(Chromosome kromosom)**: method ini untuk merubah nilai gen dalam chromosome. Method ini tidak mengembalikan value (void).
* **crossover(Chromosome parent1, Chromosome parent2)**: method ini untuk mengawinkan dua chromosome yang akan menghasilkan dua chromosome baru. Tipe kembalian dari method ini adalah array of Chromosome.
* **selection()**: method ini berfungsi sebagai seleksi Chromosome menggunakkan metode roulette wheel. Tipe kembalian dari method ini adalah Chromosome.
* **fitness(Chromosome kromosom)**: method ini untuk menghitung score chromosome. Method ini tidak mengembalikan value (void).
* **sortFitness()**: method ini untuk mengurutkan chromosome dari score fitness terbesar ke yang terkecil. Method ini tidak mengembalikan value (void).
* **findLocation()**: method ini untuk mencari posisi blank tile ada di index berapa. Tipe kembalian dari method ini adalah integer.
* **compareInitialGoal(int[] current, int[] goal)**: method ini membandingkan berapa banyak kesamaan antara array state saat ini (current) dan array goal. Tipe kembalian dari method ini adalah integer.
* **fitnessPopulation()**: method ini untuk menghitung skor populasi yang membantu perhitungan di method selection().
* **Kelas Process**

Kelas Process berguna untuk menjalankan N-Puzzle yang akan memanggil method-method dari kelas Population dan GeneticAlgorithm. Kelas ini akan mengembalikan solusi terbaik untuk menyelesaikan N-Puzzle tersebut.