**Tugas Kecil 1 IF2211 Strategi Algoritma**

**Penyelesaian IQ Puzzler Pro dengan Algoritma Brute Force**



Disusun Oleh:

Ahmad Wicaksono (13523121)

**SEKOLAH TEKNIK ELEKTRO DAN INFORMARTIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2024**

1. **LATAR BELAKANG**
   1. **IQ Puzzler Pro**

**IQ Puzzle Pro** adalah permainan teka-teki yang dirancang untuk mengasah kemampuan berpikir logis, pemecahan masalah, dan keterampilan spasial. Pemain ditantang untuk menyusun atau mencocokkan berbagai bentuk sesuai dengan aturan yang telah ditetapkan. Permainan ini memiliki tingkat kesulitan yang beragam, mulai dari level dasar hingga tingkat yang lebih kompleks. IQ Puzzle Pro tidak hanya memberikan hiburan tetapi juga berkontribusi dalam meningkatkan konsentrasi, ketelitian, dan kreativitas dalam menyelesaikan masalah. Oleh karena itu, permainan ini sering digunakan sebagai sarana edukatif maupun latihan kognitif untuk berbagai kelompok usia.

**1.2. Algoritma Brute Force**

Algoritma Brute Force merupakan metode pemecahan masalah yang dilakukan dengan mencoba semua kemungkinan solusi secara sistematis hingga menemukan hasil yang benar. Pendekatan ini tidak mengandalkan strategi optimasi khusus, melainkan mengevaluasi setiap opsi yang tersedia. Cara kerja brute force meliputi enumerasi semua kemungkinan solusi, mengevaluasi setiap solusi apakah memenuhi syarat atau tidak, serta memilih solusi terbaik jika terdapat lebih dari satu jawaban yang valid. Contoh penerapan algoritma ini dapat ditemukan dalam pencarian kata sandi dengan mencoba seluruh kombinasi karakter, pencarian pola dalam teks dengan membandingkan pola pada setiap posisi dalam teks, serta penyelesaian masalah *Traveling Salesman Problem (TSP)* dengan menghitung semua kemungkinan rute untuk menemukan yang paling optimal. Meskipun brute force memiliki keunggulan dalam kemudahan implementasi dan jaminan menemukan solusi optimal, metode ini memiliki kelemahan utama berupa ketidakefisienan, terutama pada permasalahan dengan ruang solusi yang sangat besar. Waktu eksekusi yang cenderung lama membuat brute force kurang ideal untuk masalah berskala besar, sehingga sering kali digunakan sebagai pendekatan dasar sebelum menerapkan algoritma yang lebih efisien seperti *greedy*, *dynamic programming*, atau metode heuristik.

**1.3. Penyelesaian IQ Puzzler Pro Menggunakan Algoritma Brute Force**

Dalam menyelesaikan IQ Puzzle Pro, algoritma brute force dengan backtracking digunakan untuk mencoba semua kemungkinan penyusunan bentuk secara sistematis, sambil menghindari jalur yang tidak menghasilkan solusi valid. Algoritma ini bekerja dengan menempatkan setiap bagian puzzle di posisi yang memungkinkan dan memeriksa apakah susunan tersebut sesuai dengan aturan permainan. Jika suatu konfigurasi tidak valid, algoritma akan kembali ke langkah sebelumnya (backtracking) dan mencoba alternatif lain tanpa harus mengeksplorasi seluruh kemungkinan secara menyeluruh. Dengan cara ini, banyak percobaan yang tidak perlu dapat dieliminasi lebih awal, sehingga mengurangi jumlah kemungkinan yang harus diperiksa dibandingkan brute force murni. Meskipun pendekatan ini lebih efisien daripada brute force tanpa optimasi, kompleksitasnya tetap tinggi, terutama jika ruang solusi sangat besar. Oleh karena itu, metode ini lebih cocok untuk teka-teki dengan jumlah kemungkinan yang masih dapat dikendalikan.

1. **IMPLEMETASI ALGORITMA DALAM BAHASA JAVA**

**2.1. File inputPuzzle.java**

FileinputPuzzle.java digunakan sebagai sumber data untuk membaca dan memproses grid puzzle menjadi daftar matriks biner berdasarkan karakter unik yang terdapat dalam grid.

|  |  |
| --- | --- |
| **Nama Fungsi** | **Deskripsi** |
| readInput(String filename) | Membaca file input yang berisi dimensi dan label puzzle, lalu memprosesnya menjadi list puzzle berdasarkan karakter unik yang ada |
| processPuzzle(List<String> lines, char label) | Menyaring bagian grid yang mengandung label tertentu, memangkas bagian kosong, dan mengubahnya menjadi format matriks biner untuk disimpan dalam daftar puzzle. |

**2.2. File rotation.java**

File ini digunakan untuk menghasilkan semua transformasi unik dari sebuah puzzle dalam bentuk matriks, termasuk rotasi 90 derajat searah jarum jam dan pencerminan horizontal.

|  |  |
| --- | --- |
| **Nama Fungsi** | **Deskripsi** |
| getTransformations(int[][] puzzle) | Menghasilkan daftar transformasi unik dari puzzle, termasuk empat rotasi dan pencerminan horizontal. |
| rotateRight(int[][] puzzle) | Memutar puzzle 90 derajat searah jarum jam. |
| mirror(int[][] puzzle) | Mencerminkan puzzle secara horizontal (membalik urutan kolom). |

**2.3. File bruteForce.java**

File ini digunakan untuk menyusun potongan puzzle ke dalam papan menggunakan algoritma brute force dengan mempertimbangkan semua rotasi dan pencerminan setiap potongan.

|  |  |
| --- | --- |
| **Nama Fungsi** | **Deskripsi** |
| solve(int puzzleIndex) | Mencoba menyusun semua potongan puzzle pada papan menggunakan brute force dengan rekursi dan backtracking. |
| canPlace(int[][] puzzle, int x, int y) | Memeriksa apakah potongan puzzle dapat ditempatkan di posisi (x, y) pada papan tanpa tumpang tindih. |
| placePuzzle(int[][] puzzle, int x, int y, char value) | Menempatkan atau menghapus potongan puzzle di papan dengan karakter tertentu (label atau . untuk menghapus). |
| initializeBoard() | Menginisialisasi papan dengan karakter . dan mengurutkan potongan puzzle berdasarkan ukuran (dengan yang terbesar ditempatkan lebih dulu). |
| countOnes(int[][] puzzle) | Menghitung jumlah sel bernilai 1 dalam suatu potongan puzzle untuk menentukan ukurannya. |

**2.4. File outputPuzzle.java**

File ini digunakan untuk menampilkan dan menyimpan hasil penyusunan puzzle dalam bentuk tampilan warna di terminal, gambar PNG, serta file teks.

|  |  |
| --- | --- |
| **Nama Fungsi** | **Deskripsi** |
| getAnsiColor(char letter) | Mengembalikan kode warna ANSI untuk tampilan warna-warni di terminal berdasarkan karakter. |
| getColor(char letter) | Mengembalikan warna Color untuk digunakan dalam pembuatan gambar berdasarkan karakter. |
| printBoardWithColor(char[][] board) | Menampilkan papan puzzle di terminal dengan warna sesuai karakter. |
| saveBoardAsImage(char[][] board, String filename) | Menyimpan papan puzzle sebagai gambar PNG dengan warna yang sesuai. |
| saveBoardAsText(char[][] board, String filename) | Menyimpan papan puzzle sebagai file teks dengan representasi karakter. |

**2.5 File main.java**

File ini berfungsi sebagai program utama yang membaca input puzzle dari file, memprosesnya menggunakan algoritma brute-force untuk menemukan solusi, menampilkan hasilnya di terminal, serta memberikan opsi untuk menyimpan solusi dalam bentuk teks dan gambar.

1. **SOURCE CODE DALAM BAHASA JAVA**

**3.1. Repositori Github**

**3.2. Source Code Program**

**3.2.1. inputPuzzle.java**

**3.2.2. rotation.java**

**3.2.3. bruteForce.java**

**3.2.4. outputPuzzle.java**

**3.2.5. Main.java**

1. **INPUT DAN OUTPUT PROGRAM**

**4.1. Test Case 1**

**4.2. Test Case 2**

**4.3. Test Case 3**

**4.4. Test Case 4**

**4.5. Test Case 5**

**4.6. Test Case 6**

**4.7. Test Case 7**

1. **LAMPIRAN**