

Tugas Kecil 1 IF2211 Strategi Algoritma

Semester II tahun 2024/2025

Penyelesaian IQ Puzzler Pro dengan Algoritma Brute Force



Disusun oleh:

Lukas Raja Agripa – 13523158

INSTITUT TEKNOLOGI BANDUNG

2024

Deskripsi Masalah

1.1 Permainan IQ Puzzler Pro

IQ Puzzler Pro adalah sebuah permainan puzzle yang menantang pemain untuk mengisi papan dengan berbagai potongan puzzle yang memiliki bentuk unik. Setiap potongan dapat dirotasi atau dicerminkan untuk menyesuaikan dengan ruang yang tersedia di papan. Tujuan utama dari permainan ini adalah menempatkan semua potongan ke dalam papan tanpa ada tumpang tindih dan tanpa meninggalkan ruang kosong. Permainan ini menguji kemampuan logika, strategi, dan kesabaran pemain.

1.2 Algoritma Brute Force

Algoritma brute force adalah pendekatan yang digunakan untuk menyelesaikan masalah dengan mencoba semua kemungkinan solusi secara sistematis. Dalam konteks IQ Puzzler Pro, algoritma ini mencoba semua kemungkinan penempatan potongan di papan, termasuk rotasi dan pencerminan, hingga menemukan konfigurasi yang valid. Kelebihan dari pendekatan ini adalah kesederhanaannya, karena tidak memerlukan heuristik atau optimasi khusus. Namun, kekurangannya adalah waktu eksekusi yang bisa sangat lama, terutama untuk papan yang besar dan jumlah potongan yang banyak.

Penjelasan Program

2.1 Langkah langkah program – Brute Force yang tidak heuristik

Program yang dibuat menggunakan pendekatan brute force murni, tanpa heuristik atau optimasi tambahan. Berikut adalah langkah-langkah detailnya:

a. Membaca File Input:

Program dimulai dengan membaca file input yang berisi informasi tentang papan dan potongan puzzle. File input memiliki format tertentu, yaitu:

Baris pertama: dimensi papan ($N \times M$) dan jumlah potongan (P).

Baris kedua: jenis kasus (DEFAULT/CUSTOM/PYRAMID).

Baris selanjutnya: bentuk setiap potongan, diwakili oleh karakter huruf.

b. Inisialisasi Papan dan Potongan:

Setelah membaca file input, program menginisialisasi papan sebagai grid kosong dengan ukuran yang telah ditentukan. Potongan-potongan puzzle diinisialisasi berdasarkan bentuk yang diberikan. Setiap potongan direpresentasikan sebagai daftar koordinat sel yang terisi.

c. Menempatkan Potongan:

Program mencoba menempatkan setiap potongan di setiap posisi yang mungkin di papan. Ini dilakukan dengan iterasi melalui semua sel di papan dan mencoba semua rotasi yang mungkin. Jika suatu potongan dapat ditempatkan di posisi tertentu tanpa bertabrakan dengan potongan lain, program melanjutkan ke potongan berikutnya. Jika tidak, program mencoba posisi atau rotasi lain.

d. Backtracking:

Jika penempatan suatu potongan menyebabkan tidak ada solusi yang valid untuk potongan berikutnya, program melakukan backtracking. Artinya, program menghapus potongan tersebut dan mencoba posisi atau rotasi lain. Proses ini diulang sampai semua kemungkinan penempatan potongan telah dicoba.

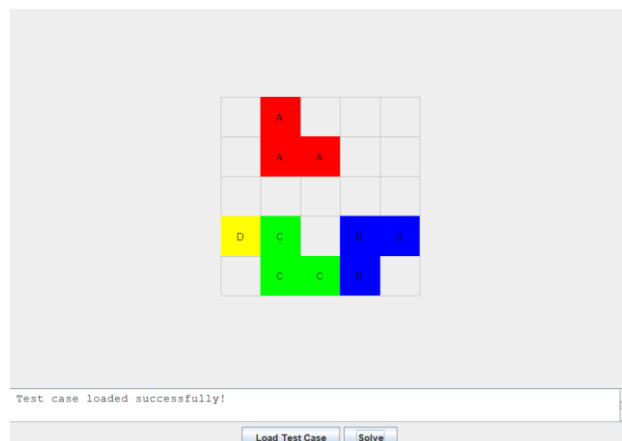
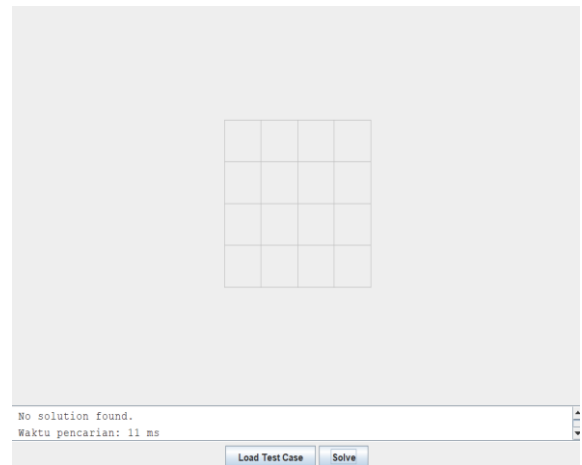
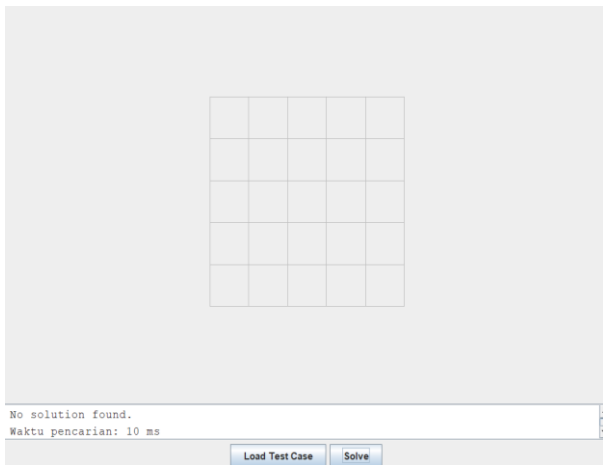
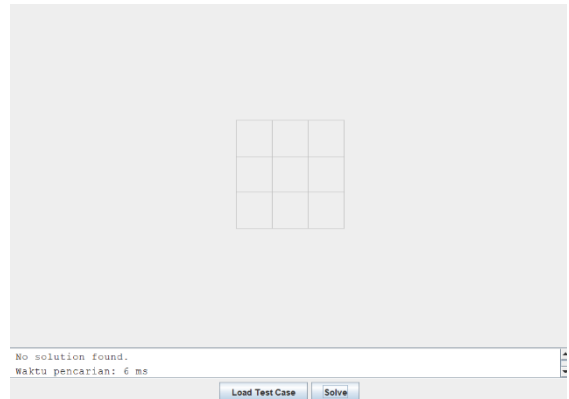
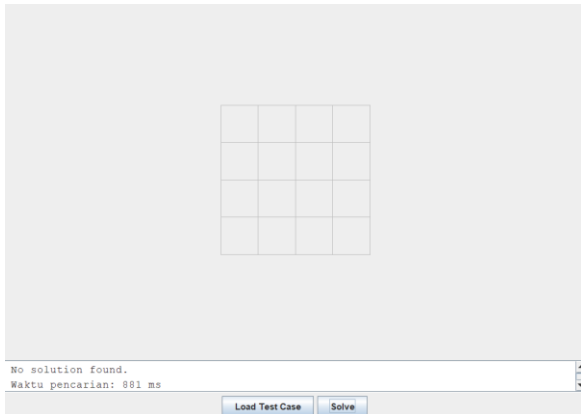
e. Mencari Solusi:

Program terus mencoba semua kemungkinan penempatan potongan sampai solusi ditemukan atau semua kemungkinan telah diuji. Jika solusi ditemukan, program menampilkan konfigurasi papan yang berhasil diisi. Jika tidak ada solusi, program menampilkan pesan "*No solution found*".

f. Menampilkan Hasil:

Setelah menemukan solusi, program menampilkan waktu pencarian dan jumlah kasus yang ditinjau.

Tangkapan layar



Pranala

Link repository : https://github.com/rlukassa/Tucil1_13523158

No	Poin	Ya	Tidak
1	Program berhasil dikompilasi tanpa kesalahan	✓	
2	Program berhasil dijalankan	✓	
3	Solusi yang diberikan program benar dan mematuhi aturan permainan		✓
4	Program dapat membaca masukan berkas .txt serta menyimpan solusi dalam berkas .txt	✓	
5	Program memiliki <i>Graphical User Interface</i> (GUI)	✓	
6	Program dapat menyimpan solusi dalam bentuk file gambar	✓	
7	Program dapat menyelesaikan kasus konfigurasi <i>custom</i>		✓
8	Program dapat menyelesaikan kasus konfigurasi Piramida (3D)		✓
9	Program dibuat oleh saya sendiri	✓	