

Tugas Kecil 1 IF2211 Strategi Algoritma
Semester II tahun 2024/2025
Penyelesaian **IQ Puzzler Pro** dengan Algoritma Brute Force

Revisi 1 : Selasa, 18 Februari 2025, pukul 13.35 WIB (Kondisi penyelesaian puzzle)

Batas pengumpulan : Hari Senin, 24 Februari 2025 pukul 13.00 WIB

Arsip pengumpulan :

- *Source* program yang dapat dijalankan disertai README
- Laporan (*soft copy*)

Deskripsi Tugas



Gambar 1 Permainan IQ Puzzler Pro
(Sumber: <https://www.smartgamesusa.com>)

IQ Puzzler Pro adalah permainan papan yang diproduksi oleh perusahaan Smart Games. Tujuan dari permainan ini adalah pemain harus dapat mengisi seluruh papan dengan piece (blok puzzle) yang telah tersedia.

Komponen penting dari permainan IQ Puzzler Pro terdiri dari:

1. **Board (Papan)** – Board merupakan komponen utama yang menjadi tujuan permainan dimana pemain harus mampu mengisi seluruh area papan menggunakan blok-blok yang telah disediakan.
2. **Blok/Piece** – Blok adalah komponen yang digunakan pemain untuk mengisi papan kosong hingga terisi penuh. Setiap blok memiliki bentuk yang unik dan semua blok harus digunakan untuk menyelesaikan puzzle.

Permainan dimulai dengan **papan yang kosong**. Pemain dapat meletakkan blok puzzle sedemikian sehingga **tidak ada blok yang bertumpang tindih** (kecuali dalam kasus 3D). Setiap blok puzzle dapat **dirotasikan** maupun **dicerminkan**. Puzzle dinyatakan **selesai** jika dan hanya jika papan **terisi penuh** dan **seluruh blok puzzle berhasil diletakkan**.

Tugas anda adalah menemukan cukup satu solusi dari permainan **IQ Puzzler Pro** dengan menggunakan **algoritma Brute Force**, atau menampilkan bahwa solusi tidak ditemukan jika tidak ada solusi yang mungkin dari puzzle.

Ilustrasi kasus

Diberikan sebuah wadah berukuran 11 x 5 serta 12 buah blok puzzle dan beberapa blok telah ditempatkan dengan bentuk sebagai berikut.



Gambar 2 Awal Permainan Game IQ Puzzler Pro
(Sumber: <https://www.smartgamesusa.com>)

Pemain berusaha untuk mengisi bagian papan yang kosong dengan menggunakan blok yang tersedia

PERHATIAN: Untuk tucil ini permainan diawali dengan papan kosong



Gambar 3 Pemain Mencoba Semua Kemungkinan
(Sumber: <https://www.smartgamesusa.com>)

Permainan dinyatakan selesai jika pemain mampu mengisi seluruh papan dengan blok (dalam tugas kecil ini ada kemungkinan pemain tidak dapat mengisi seluruh papan)

Puzzle berikut dinyatakan telah selesai karena papan sudah terisi penuh dan seluruh blok telah digunakan.



Gambar 4 Pemain Menyelesaikan Permainan
(Sumber: <https://www.smartgamesusa.com>)

Agar lebih jelas, amati video cara bermain berikut:

<https://youtube.com/shorts/MWiPAS3wfGM?feature=shared>

Spesifikasi Wajib

- Buatlah program sederhana dalam bahasa **Java** yang mengimplementasikan **algoritma Brute Force** untuk mencari solusi dalam permainan IQ Puzzler Pro.
- Algoritma brute force yang diimplementasikan harus bersifat “**murni**”, tidak boleh memanfaatkan [heuristik](#).
- Papan yang perlu diisi mulanya akan selalu kosong.
- Sebuah blok puzzle bisa saja dirotasi maupun dicerminkan sebelum diletakan pada papan.
- **Input:** program akan memberikan pengguna sebuah prompt untuk memilih file *test case* berekstensi **.txt**, kemudian program membaca file *test case* tersebut yang berisi
 1. **Dimensi Papan** terdiri atas dua buah variabel **N** dan **M** yang membentuk papan berdimensi NxM.
 2. **Banyak blok puzzle** direpresentasikan oleh variabel integer **P**.
 3. **Jenis kasus** sebuah variabel string **S** yang digunakan untuk mengidentifikasi kasus konfigurasi, hanya mungkin bernilai salah satu diantara **DEFAULT/CUSTOM/PYRAMID**.

4. **Bentuk blok puzzle** yang dilambangkan oleh konfigurasi *Character* berupa huruf. Akan ada **P** buah blok puzzle berbeda yang dibentuk oleh **P buah huruf berbeda**. *Character* yang digunakan adalah huruf **A-Z dalam kapital**.

File .txt yang akan dibaca memiliki format sebagai berikut

```
N M P
S
puzzle_1_shape
puzzle_2_shape
...
puzzle_P_shape
```

Contoh Test case

```
5 5 7
DEFAULT
A
AA
B
BB
C
CC
D
DD
EE
EE
E
FF
FF
F
GGG
```

- **Output:**

1. Tampilkan **konfigurasi blok puzzle** yang berhasil mengisi papan. **Gunakan print berwarna** untuk menunjukkan blok puzzle dengan jelas. Pastikan setiap blok puzzle berbeda memiliki warna berbeda. Beri tahu pengguna apabila puzzle tidak memiliki solusi.
2. **Waktu eksekusi** program dalam *milisecond* (tidak termasuk waktu membaca masukan dan menyimpan solusi, cukup waktu pencarian oleh algoritma).
3. **Banyak kasus** atau jumlah iterasi yang ditinjau oleh algoritma brute force.

4. **Prompt untuk menyimpan solusi** dalam sebuah berkas berekstensi **.txt** (Struktur untuk file output dibebaskan).

Contoh *output* berdasarkan contoh *Input* diatas

```
AGGGD
AABDD
CCBBF
CEEFF
EEEFF
```

Waktu pencarian: 604 ms

Banyak kasus yang ditinjau: 7387

Apakah anda ingin menyimpan solusi? (ya/tidak)

- Sangat disarankan untuk menggunakan [*semantic commit*](#).
- Berkas laporan yang dikumpulkan adalah laporan dalam bentuk PDF yang setidaknya berisi:
 1. Algoritma brute force yang digunakan, jelaskan langkah-langkahnya, **bukan hanya notasi pseudocode**.
 2. *Source code* program (pastikan bahwa program telah dapat dijalankan).
 3. Tangkapan layar yang memperlihatkan *input* dan *output*, minimal sebanyak **7** buah contoh (pastikan ada setidaknya 1 untuk masing-masing kasus bonus jika mengerjakan).
 4. Pranala ke *repository* yang berisi kode program.
- Program disimpan dalam repository yang bernama **Tucil1_NIM**. Berikut merupakan struktur dari isi repository tersebut:
 1. Folder **src** berisi *source code* program.
 2. Folder **bin** berisi *executable file* (Sesuaikan dengan bahasa pemrograman yang digunakan).
 3. Folder **test** berisi solusi jawaban dari data uji yang digunakan dalam laporan.
 4. Folder **doc** berisi laporan tugas kecil dalam bentuk PDF.
 5. **README** yang minimal berisi:
 - a. Penjelasan singkat program yang dibuat.
 - b. Requirement program dan instalasi tertentu bila ada.
 - c. Cara mengkompilasi program bila perlu dikompilasi.

- d. Cara menjalankan dan menggunakan program.
- e. Author / identitas pembuat.
- Laporan dikumpulkan hari **Senin, 24 Februari 2025** pada alamat Google Form berikut paling lambat pukul **13.00 WIB**:
<https://bit.ly/tucil1stima25>
- Pertanyaan terkait tugas kecil ini bisa disampaikan melalui QnA berikut:
<https://bit.ly/QnA-Stima-25>

Spesifikasi Bonus

Poin **maksimal** untuk spesifikasi bonus adalah **10 poin**. Silahkan pilih spesifikasi bonus yang ingin Anda kerjakan:

- **Output berupa Gambar (2 poin)**
Pengguna dapat menyimpan solusi suatu puzzle dalam bentuk file gambar. Pastikan setiap blok puzzle yang ditampilkan memiliki warna berbeda.
- **Graphical User Interface (8 poin)**
Buatlah GUI untuk program yang Anda buat. Interface ini harus memvisualisasikan papan yang sudah terisi oleh blok puzzle berwarna. Kakas untuk implementasi GUI dibebaskan.
- **Konfigurasi Custom (2 poin)**
Umunya, IQ Puzzle Pro mengharuskan Anda mengisi papan sampai penuh menggunakan seluruh blok puzzle yang tersedia. Namun, ada kalanya puzzle ini justru mengharuskan Anda untuk membentuk suatu konfigurasi tertentu walaupun papan tidak penuh. Sebagai contoh, perhatikan gambar berikut



Gambar 5 Puzzle dengan konfigurasi non persegi panjang
(Sumber: <https://www.tutete.com>)

Tugas Anda adalah untuk mengimplementasikan algoritma yang dapat menyelesaikan puzzle dengan konfigurasi *custom* pada input. Akibatnya, format input akan sedikit

berbeda dari biasanya. Variabel **N** dan **M** pada input akan melambangkan dimensi papan yang bisa memuat konfigurasi. Lalu pada baris ketiga, akan diberikan sebuah **matriks NxM** yang terdiri atas titik dan X yang akan **merepresentasikan konfigurasi custom**. Pada matriks ini, *character X* melambangkan bagian pada papan yang perlu diisi untuk membentuk konfigurasi *custom*. Agar lebih jelas, diberikan contoh input dan output berikut.

```
5 7 5
CUSTOM
...X...
.XXXXX.
XXXXXXX
.XXXXX.
...X...
A
AAA
BB
BBB
CCCC
C
D
EEE
E
```

```
  A
 CDAAA
CCCCBBB
EEEBB
 E
```

- **Konfigurasi Piramida (10 poin)**

IQ Puzzler Pro juga memiliki puzzle yang berbentuk tiga dimensi. Berbeda dengan puzzle biasanya dimana kita menyusun blok tanpa tumpang tindih, disini kita justru perlu menumpuk puzzle agar membentuk suatu konfigurasi 3D. Pada kasus ini, Anda harus membentuk konfigurasi piramida dengan alas persegi. Perhatikan bahwa pada konfigurasi ini blok puzzle bisa saja disusun dengan rotasi yang tidak umum pada kasus 2 dimensi.



Gambar 6 Puzzle dengan konfigurasi piramida
(Sumber: <https://houtenonderwijsmateriaal.be>)

Karena piramida memiliki alas persegi, input **N** dan **M** akan **bernilai sama**. Lalu, setiap lapisan pada piramida berbentuk persegi dengan panjang sisi yang bernilai 1 lebih kecil dari lapisan dibawahnya. Untuk output, visualisasikan konfigurasi 3D dengan menampilkan masing-masing lapisan dimulai dari lapisan paling atas. Contoh input dan output untuk konfigurasi piramida adalah sebagai berikut

```
3 3 6
PYRAMID
A
AA
BB
CC
DDD
E
EE
F
```

A
BC
AF
BCD
EAD
EED

Perhatian

- **Dilarang keras** *copy paste* program dari internet, AI, repository lain, ataupun program milik teman. Program harus dibuat sendiri, kecurangan akan diberikan sanksi berat yaitu nilai tugas menjadi **nol**.
- Pastikan program **dapat dikompilasi setidaknya** pada *windows* dan *linux*.
- Apabila program **tidak dapat dijalankan** maka tidak akan dinilai oleh asisten.
- Tugas dikerjakan secara **individu**.
- Tambahkan tabel berikut yang diisi *checklist* (✓) pada bagian lampiran laporan Anda.

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		

--- *Selamat Mengerjakan!* ---

“Loh mirip yang sering kulihat di reels IG”

--- Asisten ---