

Penyelesaian Jigsaw Puzzle dengan Algoritma Branch and Bound dan Brute Force

Hanifah Azhar 13509016
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509016@std.stei.itb.ac.id

Abstract—Jigsaw Puzzle merupakan sebuah permainan yang cukup sederhana. Dalam permainan ini, ada sebuah gambar yang telah dipotong menjadi bagian-bagian yang disebut puzzle pieces. Pada awal permainan, posisi puzzle pieces acak. Jigsaw Puzzle selesai ketika setiap puzzle piece berada pada posisi yang tepat, yaitu membentuk gambar yang utuh. Untuk mencapai kondisi selesai setiap puzzle piece harus dicocokkan dengan puzzle piece lainnya agar mendapatkan pasangan yang tepat untuk setiap piece. Masing-masing piece memiliki lebih dari satu pasangan, namun untuk mempercepat proses pencocokan setiap piece hanya perlu menemukan satu pasangan (pasangan dari satu sisi saja). Untuk memastikan kecocokannya, ukuran dan bentuk outdent pada satu puzzle piece harus pas ukurannya dengan indent puzzle piece pasangannya.

Algoritma Branch and Bound dan Brute Force merupakan dua dari berbagai cara yang dapat digunakan untuk menghasilkan solusi jigsaw puzzle. Branch and Bound akan mencari solusi secara melebar dengan pohon simpul. Setiap simpul adalah sebuah puzzle piece. Brute Force akan mencari solusi dengan membandingkan setiap piece dengan setiap piece lainnya.

Kata kunci—Branch and Bound, Brute force, indent, jigsaw puzzle, outdent, puzzle piece.

I. PENDAHULUAN

Jigsaw puzzle adalah permainan yang cukup sederhana. Pada umumnya jigsaw puzzle berupa sebuah gambar diatas karton, yang sudah dibelah-belah sedemikian rupa sehingga menghasilkan bagian-bagian yang dapat disebut puzzle piece. Tujuan dari permainan ini adalah menggabungkan semua puzzle piece agar membentuk suatu gambar yang utuh kembali. Suatu jigsaw puzzle dapat memiliki 2 sampai lebih dari 3000 puzzle pieces, tergantung tingkat kesulitan puzzle tersebut. Sebagai contoh, jigsaw puzzle berukuran A3 dan memiliki 1000 puzzle piece dapat dikategorikan sebagai puzzle dengan tingkat kesulitan yang tinggi.

Kini, jigsaw puzzle memiliki bentuk yang sangat beragam. Tidak lagi hanya dalam bentuk dua dimensi, ada juga jigsaw puzzle yang tiga dimensi, dan ada yang dibuat sebagai sebuah game pada komputer. Versi terbaru dari jigsaw puzzle adalah sebuah jigsaw puzzle solver. Jigsaw puzzle solver yang sudah ada, menggunakan

algoritma yang berbeda-beda. Masalah jigsaw puzzle dikategorikan kedalam NP-complete.



Gambar 1.1 Contoh jigsaw puzzle 2D



Gambar 1.2 Contoh jigsaw puzzle 3D



Gambar 1.3 Contoh jigsaw puzzle pada game komputer

Pada komputer, *puzzle piece* perlu discan atau dibuat *puzzle piece* virtual. Programmer harus bisa mengimplementasikan *Computer Vision* dan *Pattern Recognition*. Agar *jigsaw puzzle* dapat terselesaikan, masing-masing *puzzle piece* harus diletakkan disamping *puzzle piece* lain yang sesuai. Untuk memastikan kecocokannya, ukuran dan bentuk “*outdent*” pada satu *puzzle piece* harus pas ukurannya dengan “*indent*” *puzzle piece* pasangannya. Dalam makalah ini, dibahas penyelesaian *jigsaw puzzle* dengan algoritma *Branch and Bound* dan *Brute Force*.

II. METODE

A. Algoritma Branch and Bound

Algoritma *Branch and Bound* melakukan pencarian secara melebar dengan menggunakan pohon simpul. Simpul akar adalah simpul pertama yang dibuat. Lalu, dibuat semua simpul anak dan simpul-simpul dari setiap simpul anak sampai habis atau kondisi terpenuhi.

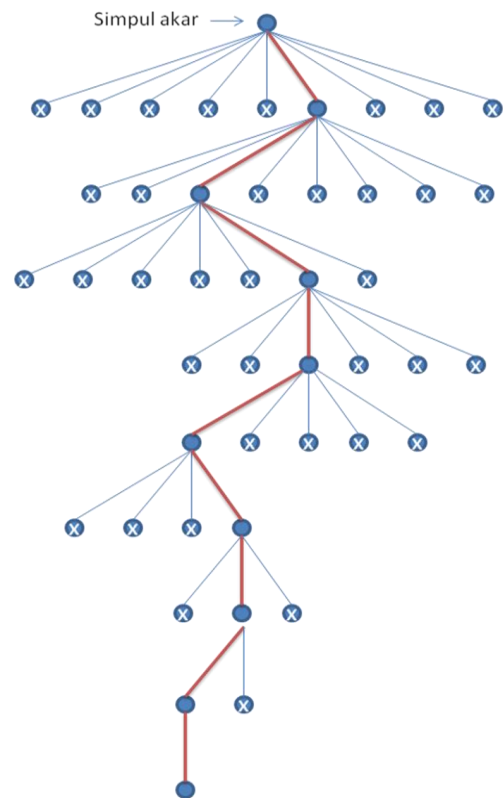
Algoritma *Branch and Bound* mirip dengan algoritma BFS. Namun, pada algoritma *Branch and Bound* pencarian ke solusi dapat dipercepat dengan memilih simpul yang masih bisa dijalankan sesuai dengan fungsi pembatas (*bound*). Pada pencarian solusi *jigsaw puzzle*, fungsi pembatas untuk *Branch and Bound* adalah kecocokan *piece* dengan *piece* lainnya. Jadi, bila *piece* anak tidak cocok dengan *piece* “*parent*”nya, simpul yang memegang *piece* anak langsung dipotong dan tidak dilanjutkan lagi.

Langkah-langkah umum yang digunakan dalam algoritma *Branch and Bound* adalah sebagai berikut:

1. Mulai dari simpul akar.
2. Periksa apakah simpul merupakan solusi.
3. Jika solusi, pencarian selesai dan hasil dikembalikan.
4. Jika fungsi keluar dari fungsi pembatas, simpul dipotong dan pencarian lanjut ke simpul di berikutnya.
5. Jika fungsi keluar tidak dari fungsi pembatas namun belum solusi, lanjutkan ke semua simpul anak (yang bertetangga dengan simpul tersebut).
6. Ulangi pencarian dari langkah kedua.
7. Jika semua simpul sudah diperiksa, pencarian selesai dan mengembalikan hasil (hasil yang dikembalikan berupa: selesai dan solusi terbaik yang didapat atau tidak selesai).

Penjelasan untuk Gambar 2.1:

- Simpul paling atas adalah simpul akar.
- Simpul yang diberi tanda x memiliki arti bahwa *piece* tersebut tidak cocok dengan *piece* pada simpul *parent*-nya.
- Jalur yang diberi warna merah adalah jalur penelusuran yang menghasilkan solusi.



Gambar 2.1 Penelusuran Branch and Bound untuk 10 puzzle pieces

Jumlah perbandingan yang dilakukan pada Branch and Bound untuk *jigsaw puzzle* dengan $n = 10$ (n = jumlah *piece*) adalah :

$$\begin{aligned} \text{Jumlah perbandingan} &= 9+8+7+6+5+4+3+2 \text{ (piece terakhir tidak perlu dibandingkan karena dianggap sudah pasti benar)} \\ &= 44 \end{aligned}$$

Rumus yang didapat untuk menghitung jumlah perbandingan yang dilakukan dengan *piece* sejumlah n adalah :

$$F(n) = (n-1) + (n-2) + (n-3) + \dots + \dots + (n - (n-2))$$

$F(n)$: Jumlah perbandingan untuk *piece* sebanyak n (*piece* terakhir tidak perlu dibandingkan karena dianggap sudah pasti benar)

B. Algoritma Brute Force

Brute force adalah suatu cara penyelesaian masalah dengan sangat sederhana, yaitu dengan cara yang sudah *obvious*. Algoritma *brute force* tidak mementingkan kemangkusan ataupun waktu penyelesaian. Penggunaan algoritma *Brute Force* sangat beragam dan dapat digunakan untuk memecahkan sebagian besar masalah yang ada.

Pada *jigsaw puzzle*, metode paling *obvious* untuk dilaksanakan adalah membandingkan setiap *piece* dengan setiap *piece* lainnya.

Dengan mengasumsikan setiap *piece* yang dicari ada pada *piece* terakhir diperiksa (kasus terburuk) didapat rumus sebagai berikut:

$$F(n) = (n) + (n-1) + (n-2) + \dots + \dots + (n - (n-2))$$

$F(n)$: Jumlah perbandingan untuk *piece* sebanyak n (*piece* terakhir tidak perlu dibandingkan karena dianggap sudah pasti benar)

Namun, bila mengasumsikan setiap *piece* yang dicari ada pada *piece* pertama yang diperiksa (kasus terbaik) didapat rumus sebagai berikut:

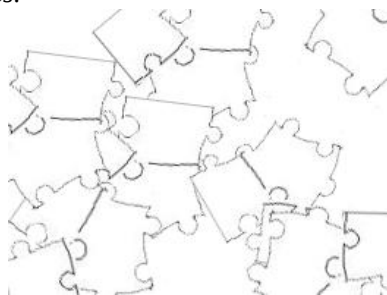
$$F(n) = (n-1)$$

$F(n)$: Jumlah perbandingan untuk *piece* sebanyak n (*piece* terakhir tidak perlu dibandingkan karena dianggap sudah pasti benar)

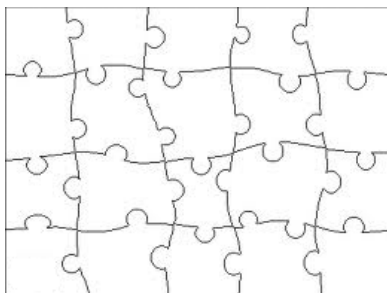
C. Penyelesaian Jigsaw Puzzle dengan Algoritma Branch and Bound

Pencarian pada *jigsaw puzzle* dimulai dari simpul akar, yaitu *puzzle piece* pertama yang didapat.. Lalu, membuat simpul-simpul lainnya yang merupakan *puzzle piece* berikutnya yang akan dicocokkan.

Sebagai contoh, akan digunakan *jigsaw puzzle* berisi 20 *puzzle pieces*.



Gambar 2.3 Kondisi awal

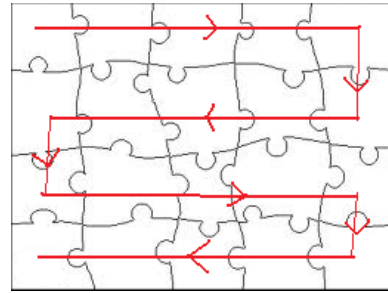


Gambar 2.4 Kondisi yang ingin dicapai

Langkah-langkah:

1. Kondisi awal adalah posisi semua *puzzle piece* masih acak.
2. Semua *piece* diberi nomer dari satu hingga 20.
3. *Piece* nomer satu dijadikan simpul akar.

4. Mulai dibuat simpul anak dari *piece* dua hingga 20.
5. Pencocokan *piece* dengan arah seperti gambar dibawah ini:



Gambar 2.5 Matriks arah pencocokan Branch and Bound

Mulai dari baris 1 kolom 1 (paling kiri), kolom terus bertambah 1. Ketika sudah sampai kolom 5 (paling kanan), baris bertambah 1. Proses dilanjutkan hingga ke baris 4 kolom 5.

Contoh implementasi: bila *puzzle piece* pada simpul akar tidak memiliki sisi kanan yang rata maka simpul anak akan diperiksa kecocokannya dengan bagian kanan *puzzle piece*. Jika sisi kanan rata, akan di periksa apakah sisi bawah rata, jika tidak maka simpul anak akan di periksa kecocokannya dengan bagian bawah *puzzle piece*. Jika dua sisi dari *piece* rata dan sudah ditemukan semua pasangannya artinya sudah sampai ujung *jigsaw puzzle*. Kemudian, kembali ke simpul akar dan memulai pencocokan dengan arah sebaliknya hingga mencapai *puzzle piece* yang memiliki sisi kiri dan atas rata.

6. Simpul diperiksa kecocokannya, bila tidak cocok maka simpul tersebut tidak akan dilanjutkan. *Piece* dikatakan cocok ketika salah satu sisi dari kedua *pieces* dapat dipasang.
7. Langkah nomer 6 dilanjutkan hingga semua simpul cocok dan tercapai kondisi akhir yang diinginkan.

Langkah-langkah diatas dapat digunakan untuk menyelesaikan *jigsaw puzzle* dengan jumlah pieces n dengan mengganti angka 20 pada langkah-langkah diatas menjadi n .

D. Penyelesaian Jigsaw Puzzle dengan Algoritma Brute-Force

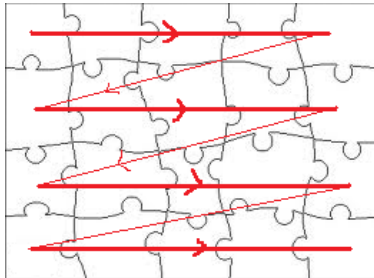
Kasus yang digunakan sama dengan kasus pada penyelesaian *jigsaw puzzle* dengan algoritma Branch and Bound. Pencarian pada *jigsaw puzzle* dimulai dari simpul akar, yaitu *puzzle piece* yang seharusnya memiliki posisi dipojok kiri atas karena hanya memiliki dua tetangga. Lalu, membuat simpul-simpul lainnya yang merupakan *puzzle piece* berikutnya yang akan dicocokkan.

Langkah-langkah:

1. Kondisi awal adalah posisi semua *puzzle piece* masih acak.
2. Melakukan pencarian untuk *piece* paling kiri-atas

(piece dengan sisi kiri dan sisi atas tidak memiliki *indent* dan *outdent*), dengan cara iterasi dari *piece* pertama yang ditemukan hingga *piece* ditemukan, lalu *piece* tersebut diberi nomor satu.

3. *Pieces* berikutnya dinomori dari dua hingga 20.
4. Memeriksa *piece* dengan iterasi dari nomor dua hingga ditemukan *piece* yang cocok untuk menjadi *next piece* sesuai dengan arah pada gambar dibawah ini



Gambar 2.6 Matriks arah pencocokan *Brute Force*

5. Mengulang langkah nomer 4 sampai kondisi terakhir terpenuhi (*puzzle* selesai)

Langkah-langkah diatas dapat digunakan untuk menyelesaikan *jigsaw puzzle* dengan jumlah *pieces* n dengan mengganti angka 20 pada langkah-langkah diatas menjadi n .

III. ANALISIS

Waktu yang dibutuhkan untuk menyelesaikan *jigsaw puzzle* dengan sebuah proses didalam komputer tidak hanya dipengaruhi oleh pengaplikasian *Computer Vision* dan *Pattern Recognition*, namun juga sangat dipengaruhi oleh algoritma yang digunakan dan jumlah perbandingan yang dilakukan. Semakin sedikit perbandingan yang dilakukan, semakin sedikit pula pemeriksaan dan pembacaan gambar yang perlu dilakukan, sehingga semakin sedikit pula waktu yang dibutuhkan untuk menyelesaikan *jigsaw puzzle*.

Berikut ini, dibahas hasil analisis dari algoritma *Branch and Bound*, *Brute Force*, dan hasil perbandingan kedua algoritma tersebut.

A. Algoritma *Branch and Bound*

Pencarian solusi permainan *Jigsaw Puzzle* menggunakan algoritma *Branch and Bound* mempunyai performansi yang baik. Penggunaan algoritma ini akan selalu menghasilkan jumlah langkah-langkah solusi yang pasti dan tetap.

B. Algoritma *Brute Force*

Pencarian solusi permainan *Jigsaw Puzzle* menggunakan algoritma *Brute Force* mempunyai rata-rata performansi yang baik. Penggunaan algoritma ini dapat menghasilkan jumlah langkah-langkah solusi yang sangat baik. Namun, pada kasus terburuk jumlah langkah-langkah solusi yang dihasilkan tidak terlalu baik.

C. Perbandingan Algoritma *Branch and Bound* dan Algoritma *Brute Force*

Pada kasus terburuk, yaitu saat setiap *piece* yang dicari selalu ditemukan pada akhir setiap tahap pencarian. Perbedaan jumlah perbandingan yang dilakukan *Branch and Bound* tidak terlalu signifikan dengan jumlah perbandingan yang dilakukan *Brute Force*. Sehingga waktu yang diperlukan untuk melaksanakan kedua algoritma juga tidak berbeda jauh.

$n = 20$			
Algoritma	Rumus	Perbandingan yang dilakukan	Total
<i>Branch and Bound</i>	$F(n) = (n-1) + (n-2) + (n-3) + \dots + (n - (n-2))$	$19+18+17+16+15+14+13+12+11+10+9+8+7+6+5+4+3+2$	189
<i>Brute Force</i>	$F(n) = (n) + (n-1) + (n-2) + \dots + (n - (n-2))$	$20+19+18+17+16+15+14+13+12+11+10+9+8+7+6+5+4+3+2$	209

Tabel 3.1 Kasus terburuk

Pada kasus terbaik, yaitu saat setiap *piece* yang dicari selalu ditemukan pada awal setiap tahap pencarian. Perbedaan jumlah perbandingan yang dilakukan *Branch and Bound* sangat signifikan dengan jumlah perbandingan yang dilakukan *Brute Force*. Sehingga untuk kasus ini, waktu yang diperlukan untuk melakukan pencarian dengan *Branch and Bound* lebih lama daripada melakukan pencarian dengan *Brute Force*.

$n = 20$			
Algoritma	Rumus	Perbandingan yang dilakukan	Total
<i>Branch and Bound</i>	$F(n) = (n-1) + (n-2) + (n-3) + \dots + (n - (n-2))$	$19+18+17+16+15+14+13+12+11+10+9+8+7+6+5+4+3+2$	189
<i>Brute Force</i>	$F(n) = (n-1)$	19	19

Tabel 3.2 Kasus terbaik

IV. KESIMPULAN

Rekor dunia untuk *jigsaw puzzle solver* pada tahun 2010 adalah 400 *pieces* selama 3 menit, dan rekor sebelumnya pada tahun 2008 adalah 320 *pieces*. Dapat disimpulkan bahwa waktu merupakan aspek yang sangat penting dalam pencarian hasil.

Permainan Jigsaw Puzzle dapat diselesaikan dengan berbagai algoritma, beberapa diantaranya adalah *Branch and Bound* dan *Brute Force*. Waktu yang dibutuhkan oleh algoritma *Branch and Bound* konstan, sehingga dapat diandalkan. Waktu yang dibutuhkan oleh algoritma *Brute Force* bervariasi. *Brute Force* bisa lebih lambat, namun peluang untuk mendapat hasil lebih cepat daripada pencarian solusi oleh *Branch and Bound* lebih besar. Namun, ketika mendapat kasus terburuk pada suatu *jigsaw puzzle* dengan jumlah *piece* sangat besar, *Brute Force* lebih lama dari *Branch and Bound*, dan perbedaan waktu cukup signifikan.

REFERENSI

- [1] Munir, Rinaldi, "Diktat Kuliah IF3051 Strategi Algoritma", Program Studi Teknik Informatika, 200
- [2] http://en.wikipedia.org/wiki/Jigsaw_puzzle
Waktu akses : Rabu, 7 Desember 2011 pukul 20.00
- [3] <http://www.puzzlehistory.com/>
Waktu akses : Rabu, 7 Desember 2011 pukul 20.02
- [4] http://www.parc.com/content/attachments/global_approach_autom_4365_parc.pdf
Waktu akses : Rabu, 7 Desember 2011 pukul 20.10
- [5] <http://www.switched.com/2010/05/23/jigsaw-puzzle-solving-computer-makes-us-feel-inadequate/>
Waktu akses : Kamis, 8 Desember 2011 pukul 6.30
- [6] http://www.webopedia.com/TERM/B/brute_force.html
Waktu akses : Kamis, 8 Desember 2011 pukul 6.31
- [7] <http://www.scientificamerican.com/article.cfm?id=solving-a-digital-jigsaw>
Waktu akses : Kamis, 8 Desember 2011 pukul 6.35
- [8] <http://www.newscientist.com/article/dn18947-computer-solves-400piece-jigsaw-to-claim-world-record.html>
Waktu akses : Kamis, 8 Desember 2011 pukul 12.30

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 December 2011



Hanifah Azhar 13509016