

# Optimasi Permainan “Tetris” dengan Pendekatan Algoritma Greedy dan Algoritma Brute Force

Ridwan<sup>1</sup>, Brahmasta Adipradana<sup>2</sup>, Weno Adji Syahdana<sup>3</sup>

Laboratorium Ilmu dan Rekayasa Komputasi  
Departemen Teknik Informatika, Institut Teknologi Bandung  
Jl. Ganesha 10, Bandung

E-mail : [if13072@students.if.itb.ac.id](mailto:if13072@students.if.itb.ac.id)<sup>1</sup>,  
[if13082@students.if.itb.ac.id](mailto:if13082@students.if.itb.ac.id)<sup>2</sup>, [if13084@students.if.itb.ac.id](mailto:if13084@students.if.itb.ac.id)<sup>3</sup>

## Abstrak

Dalam kehidupan kita sekarang ini, keberadaan game menjadi suatu bagian penting bagi kita. Kita memerlukan game, bukan hanya sekedar sebagai pengisi waktu senggang di sela-sela kesibukan kita, tapi juga untuk menjadi model dari permasalahan kehidupan kita sehari-hari. Salah satu permainan yang telah digemari sejak lama (dikeluarkan tahun 1984) sampai sekarang adalah permainan tetris. Sampai saat ini kebanyakan orang memainkan tetris hanya dengan menggunakan intuisi dan perasaan untuk menyusun balok-balok tetris yang turun. Dalam makalah ini, kami ingin menjabarkan sebuah algoritma untuk mencari solusi yang paling mendekati solusi optimum dalam permainan tetris. Algoritma yang akan kami gunakan adalah algoritma yang menggabungkan konsep-konsep algoritma Greedy dan Algoritma BruteForce.

**Kata kunci:** makalah, tetris, Greedy, BruteForce

## 1. Pendahuluan

Sampai saat ini kebanyakan orang memainkan tetris hanya dengan menggunakan intuisi dan perasaan untuk menyusun balok-balok tetris yang turun. Penyusunan balok ini harus dilakukan sedemikian rupa sehingga dapat menghasilkan nilai sebanyak mungkin, dan menjaga kelangsungan permainan selama mungkin. Dalam makalah ini, kami ingin memaparkan sebuah ide algoritma untuk mencari solusi yang mendekati solusi optimum dalam permainan tetris. Sebagai pendekatan dalam memperoleh keoptimuman digunakan algoritma Greedy, sementara itu algoritma Brute Force dipakai untuk mengevaluasi kemungkinan penumpukan balok tetris yang mungkin terjadi.

## 2. Tetris

Tetris adalah sebuah tipe permainan yang dibuat oleh sepasang *programmer* berkebangsaan Uni Soviet (sekarang Rusia). Dalam permainan tetris, balok-balok tetris berjatuh ke area permainan dalam waktu konstan. Balok tetris selalu terdiri dari 4 balok kecil yang membentuk tujuh macam rupa yang unik (berbeda satu sama lain).

Pemain dapat mengontrol balok tetris yang jatuh melalui 4 jenis kendali, yaitu 2 tombol, masing-masing arah kiri dan kanan untuk menggerakkan balok ke kiri atau ke kanan. Dua kendali yang lain adalah tombol untuk menambah kecepatan jatuhnya balok, dan satu tombol lagi untuk memutar balok 90° searah jarum jam (mengubah bentuk balok). Tujuan

permainan ini adalah untuk mengarahkan balok yang jatuh ke arah susunan tumpukan balok-balok yang telah jatuh sebelumnya., sehingga balok-balok tersebut dapat tersusun membentuk baris penuh. Setiap baris yang penuh akan dihilangkan dari tumpukan dan tumpukan di atasnya akan jatuh sesuai jumlah baris yang hilang. Jika susunan tumpukan balok mencapai garis batas atas, maka permainan berakhir.

## 3. Algoritma

Algoritma yang kami gunakan untuk mencari solusi dari permainan tetris dalam makalah ini adalah algoritma yang menggunakan konsep-konsep yang ada dalam algoritma Greedy dan Algoritma BruteForce.

Algoritma Greedy merupakan metode yang paling umum digunakan untuk memecahkan masalah optimasi. Algoritma ini sederhana dan lempang. Algoritma Greedy memecahkan masalah langkah per langkah, pada setiap langkah:

1. mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (prinsip “*take what you can get now!*”)
2. berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global

*Brute force* adalah sebuah pendekatan yang lempang (*straightforward*) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah

(*problem statement*) dan definisi konsep yang dilibatkan.

Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*).

Algoritma yang kami gunakan untuk mendapatkan susunan tumpukan balok yang paling baik dengan menempatkan balok ke tempat yang tepat. Algoritma ini menggunakan prinsip Greedy dalam mencari langkah solusi yang paling menguntungkan. Prioritas keuntungan yang kami susun terdiri dari:

1. Membentuk satu atau lebih baris paling penuh
2. Membentuk satu atau lebih baris paling mendekati penuh
3. Tidak membentuk ruang kosong pada susunan tumpukan balok
4. Balok dapat masuk ke dalam susunan tumpukan balok paling dalam

Algoritma yang kami kemukakan akan mencari penempatan balok yang jatuh ke ruang yang paling tepat sesuai prioritas keuntungan di atas diantara susunan tumpukan balok. Pencarian ini akan dilakukan secara *Brute Force*.

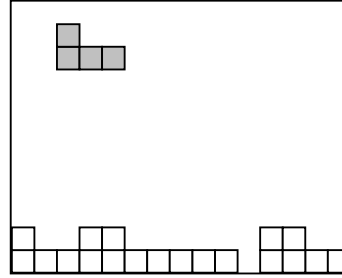
Balok yang jatuh akan dicoba untuk ditempatkan ke ruang di antara susunan tumpukan balok dibawah. Algoritma ini akan mencari penempatan yang sesuai dengan prioritas di atas. Pencarian solusi diantara susunan tumpukan balok akan dilakukan secara *Brute Force*. Algoritma ini akan mencari solusi paling menguntungkan untuk setiap sisi balok yang sedang jatuh. Pencarian solusi untuk setiap sisi dilakukan secara *Brute Force*.

Apabila pada skala prioritas tertinggi memiliki lebih dari satu solusi terbaik yang sama, maka diantara solusi tersebut akan dibandingkan satu sama lain untuk mencari yang paling menguntungkan dengan standard prioritas selanjutnya, dan begitu selanjutnya.

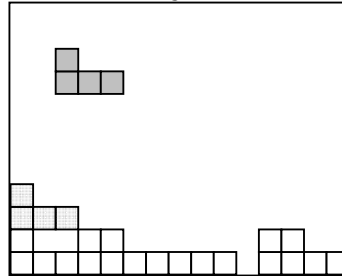
Apabila pada skala prioritas tertinggi tidak memiliki solusi, maka akan mencari solusi paling menguntungkan dengan skala prioritas selanjutnya, dan begitu selanjutnya.

Apabila pada skala prioritas tertinggi hanya memiliki satu solusi paling menguntungkan, maka akan dibandingkan dengan solusi dari hasil pencarian solusi untuk sisi balok yang lain. Diantara setiap solusi sisi balok, dicari solusi yang paling menguntungkan sesuai skala prioritas di atas. Dan balok akan ditempatkan pada ruang tersebut.

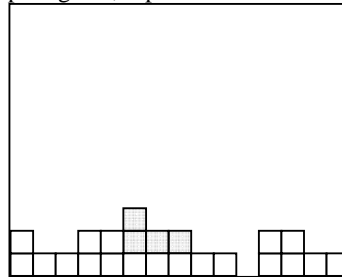
#### 4. Studi Kasus



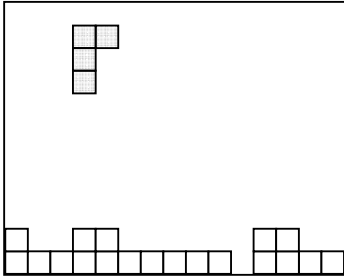
Apabila ada kasus seperti diatas, maka algoritma kami akan mencari solusi yang paling menguntungkan untuk menempatkan balok tersebut ke ruang di antara susunan tumpukan balok. Pencarian dicari secara brute force dari kiri ke kanan untuk sisi yang pertama kali keluar. Dapat dilihat seperti gambar berikut, bahwa lgoritma seakan-akan menempatkan balok tersebut dari kiri ke kanan untuk balok dengan sisi tersebut.



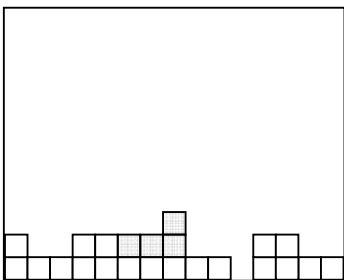
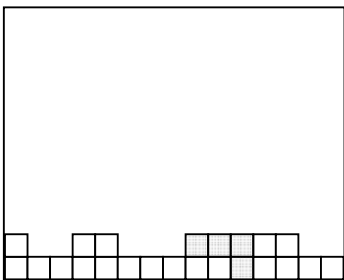
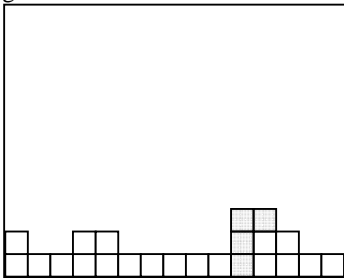
Setelah algoritma ini mencari solusi sampai paling kanan, maka algoritma ini akan menyimpan satu solusi terbaik yang ada. Apabila ada beberapa solusi yang sama baiknya, maka akan diambil penempatan paling kiri, seperti dilihat dibawah ini.



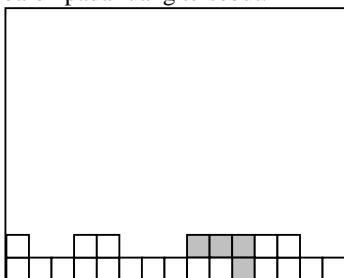
Setelah menyimpan solusi terbaik untuk sisi tersebut, maka algoritma ini akan mulai mencari solusi optimum untuk sisi berikutnya. Tampak di bawah, algoritma ini seakan-akan memutar balok untuk memulai pencarian sisi berikutnya. Sisi berikut yang kami maksud disini adalah sisi dimana balok yang sedang jatuh diputar  $90^0$  searah jarum jam.



Setelah itu, algoritma ini akan menyimpan solusi dari setiap sisi berikutnya, seperti terlihat pada tiga gambar berikut ini.



Diantara setiap solusi sisi balok, dicari solusi yang paling menguntungkan sesuai skala prioritas di atas. Dan balok akan ditempatkan pada ruang tersebut. Seperti terlihat pada gambar berikut, algoritma ini telah menemukan solusi terbaik, dan menempatkan balok pada ruang tersebut.



- Pemberian tugas makalah: **10 Mei 2005**
- Pengumpulan: **20 Mei 2005**
- Upload makalah ke situs web: **23 Mei 2005**

#### 4. Simpulan

- Solusi optimasi permainan tetris yang kami tawarkan baru sebatas konsep pemecahan dan belum diwujudkan dalam suatu notasi algoritmik. Solusi ini juga belum dapat diukur keoptimalannya karena belum dibuat perbandingan dengan algoritma lain.
- Nilai maksimum dalam permainan tetris tidak dapat dicapai dengan algoritma yang sederhana, karena membutuhkan pertimbangan yang cukup banyak dan kompleks dalam memilih jalan yang harus diambil (dalam hal ini cara menumpuk balok).
- Pemecahan dengan Algoritma Greedy dan Brute Force pada pendekatan optimasi permainan tetris tentu akan memakan sumber daya cukup banyak dan tidak mangkus, karena setiap kombinasi tumpukan harus diperiksa. Ditambah lagi untuk setiap balok tetris bisa mempunyai empat bentuk yang berbeda. Maka besar kombinasi seluruhnya akan dikalikan lagi dengan 4.
- Kami hanya mencoba memecahkan solusi melalui satu pendekatan saja, dengan prioritas yang kami tentukan sendiri. Masih banyak pendekatan lainnya dalam membuat pertimbangan langkah, terutama dalam memilih urutan prioritas dan kendala.

#### 5. Daftar Pustaka

1. Rinaldi Munir. 2005. "*Strategi Algoritmik*". Laboratorium Ilmu dan Rekayasa Komputasi – Institut Teknologi Bandung.
2. Marlo Steed. 1992. *Strategy Construction in Problem Solving with Tetris*. <http://www.uleth.ca/edu/faculty/members/steed/Articles/tetris.pdf>. Diakses tanggal 20 Mei 2005