

# Penerapan Algoritma Greedy dalam Permainan Tetris

Mario Orlando Teng / 13510057  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
mario.orlando@students.itb.ac.id

**Abstrak** — Tetris adalah salah satu permainan yang cukup populer di tahun 1990 dimana saat itu konsol-konsol game mulai bermunculan. Permainan ini bertujuan untuk menyusun sedemikian rupa potongan-potongan puzzle yang datang satu per satu menjadi satu kesatuan utuh. Di dalam makalah ini akan dibahas penerapan algoritma greedy yang telah diajarkan di dalam mata kuliah IF3051 Strategi Algoritma dalam proses penyusunan puzzle tetris ini. Algoritma greedy yang akan digunakan adalah greedy berdasarkan blok bawah dan greedy berdasarkan tingkat.

**Kata Kunci** — tetris, greedy, blok penuh, tingkatan.

## I. PENDAHULUAN

### 1.1 Tetris

Tetris diambil dari “tetra” yang dalam bahasa Yunani berarti “empat”. Tetris adalah permainan yang ditemukan oleh seorang programmer asal Uni Soviet yaitu Alexey Pajitnov pada tahun 1984. Permainan ini mulai berkembang di Uni Soviet setelah diimplementasikan di IBM PC pada tahun 1985 dan perkembangannya ini terus meningkat setelah berbagai konsol-konsol game dan PC bermunculan dan mulai memasyarakat di awal tahun 1990.

Permainan tetris sendiri adalah permainan yang bertujuan untuk menyusun potongan-potongan blok menjadi sebuah keutuhan satu baris atau lebih. Jika sebuah atau lebih baris telah terisi secara penuh maka baris tersebut akan hilang sehingga tinggi tumpukan blok akan berkurang, pemain akan kalah jika tinggi blok mencapai puncak yang telah ditentukan.

Cara bermain tetris adalah dengan melakukan kontrol terhadap blok-blok puzzle yang ada. Potongan-potongan blok akan “jatuh” satu per satu dari atas dan pemain dapat memutar blok tersebut sebesar  $90^\circ$  dengan arah searah jarum jam untuk sekali putar. Selain itu pemain dapat menggerakkan blok tersebut ke arah kiri dan kanan untuk mengarahkan blok ke arah yang diinginkan. Arah bawah juga dapat dipakai untuk menambah kecepatan “jatuh” dari blok puzzle.

Berikut adalah blok-blok puzzle yang ada di dalam permainan tetris. Blok-blok ini berbentuk seperti huruf O, I, S, Z, L, J, dan T.

## Tetris Pieces

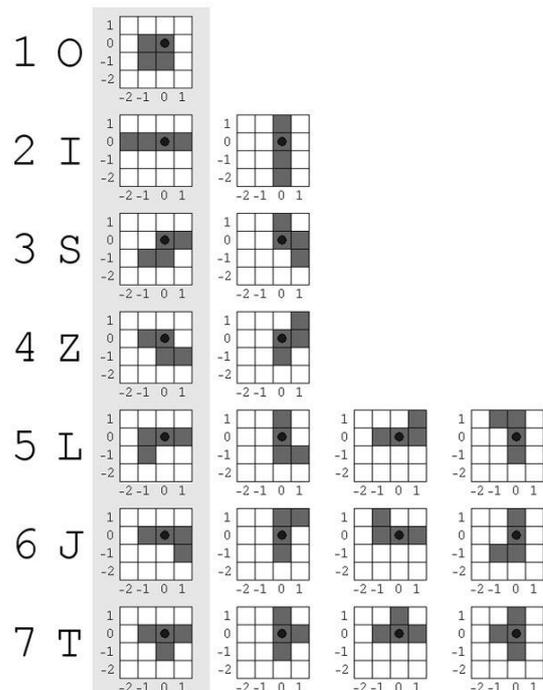


Figure 1 Blok-Blok Tetris

### 1.2 Algoritma Greedy

Dalam berbagai permasalahan yang ada terkadang diperlukan sebuah cara untuk mencari sebuah solusi optimal dari berbagai solusi yang ada untuk masalah tersebut. Algoritma yang cukup populer dalam mencari solusi optimasi adalah algoritma greedy.

Perlu diketahui terlebih dahulu bahwa algoritma greedy adalah algoritma yang membentuk solusi langkah per langkah (step by step). Sesuai dengan namanya, algoritma ini adalah algoritma yang mempertimbangkan keuntungan tertinggi dari langkah yang akan diambil berikutnya. Dengan melakukan hal ini berarti algoritma greedy mengambil langkah dengan perolehan terbaik lokal (optimum lokal) dengan harapan bahwa untuk langkah-langkah selanjutnya akan diperoleh perolehan terbaik global (optimum global).

Dari bahasan di atas dapat disimpulkan bahwa algoritma greedy adalah algoritma yang membentuk solusi langkah per langkah dengan kondisi setiap langkah adalah sebagai berikut :

- 1) Mengambil pilihan yang memberikan keuntungan yang paling besar saat itu tanpa memperhatikan konsekuensi ke depannya
- 2) Dengan mengambil langkah terbaik (optimum lokal) maka diharapkan untuk solusi terakhir akan diperoleh hasil terbaik juga (optimum global)

## II. LATAR BELAKANG MASALAH

Dalam sebuah permainan tentu akan lebih menarik jika kita dapat bermain dengan orang lain. Tetapi terkadang waktu tidak memungkinkan kita untuk bermain bersama secara terus menerus sehingga kita hanya dapat bermain sendiri. Oleh karena itu diperlukan sebuah program dimana program tersebut dapat memainkan permainan dengan tingkat kepintaran yang setara dengan manusia. Program ini biasa disebut dengan nama *inteligensia buatan*.

Untuk membuat sebuah program *inteligensia buatan* yang canggih maka dibutuhkan algoritma yang baik dan optimal dalam memecahkan masalah di sebuah permainan yang dalam bahasan kali ini permainan yang akan dibahas adalah tetris. Penulis akan mencoba mencari algoritma yang memiliki solusi optimal dengan penerapan greedy yang telah diajarkan di kuliah.

## III. PENERAPAN GREEDY SEBAGAI SOLUSI

Sebagai solusi dari persoalan penyusunan blok puzzle tetris ini akan digunakan penerapan greedy dengan acuan yang digunakan adalah greedy berdasarkan blok bawah dan greedy berdasarkan tingkat.

Sebelum masuk ke pembahasan dua solusi tersebut perlu diketahui terlebih dahulu bahwa akan ada penetapan nilai untuk setiap blok. Penetapan nilai untuk setiap blok ini dilihat dari banyaknya segmen blok yang berada di landasan bawah. Berikut adalah contoh penetapan nilai untuk I dan Z.

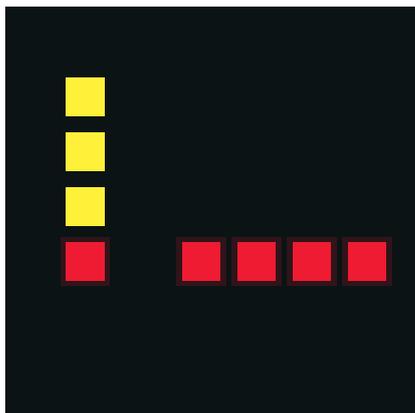


Figure 2 Penetapan Nilai Blok I

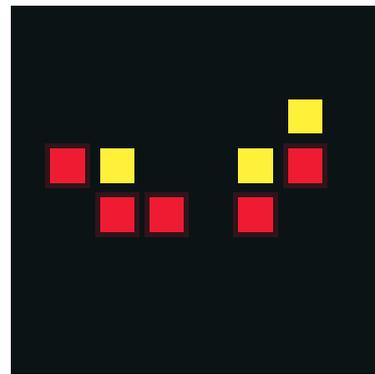


Figure 3 Penetapan Nilai Blok Z

Dari kedua gambar di atas penetapan nilai ditandai dengan bagian yang diberi warna merah. Dari sana dapat dilihat bahwa :

- 1) Blok I horizontal memiliki penetapan nilai “4”, sementara blok I vertikal memiliki penetapan nilai “3”
- 2) Blok Z horizontal memiliki penetapan nilai “3”, sementara blok Z vertikal memiliki penetapan nilai “2”

Greedy yang digunakan akan mengambil keuntungan utama berdasarkan penetapan nilai ini. Selain penetapan nilai, akan ada blok penuh dan tingkat yang mempengaruhi besarnya keuntungan.

### 3.1 Greedy Berdasarkan Blok Bawah

Blok bawah yang dimaksud di sini adalah bagian blok yang berada di bawah blok puzzle yang sedang diacu. Penuh tidaknya blok bawah tentu akan mempengaruhi keuntungan yang diambil mengingat tujuan permainan ini adalah menyusun blok-blok menjadi satu kesatuan utuh yang sepeoleh mungkin.

Untuk melakukan penghitungan keuntungan digunakan prinsip sebagai berikut. Untuk setiap blok bawah yang kosong yang diakibatkan oleh penempatan blok puzzle maka keuntungan dari penempatan blok puzzle tersebut akan dikurangi “4”. Berikut adalah contoh dari pengurangan keuntungan akibat blok bawah yang kosong.

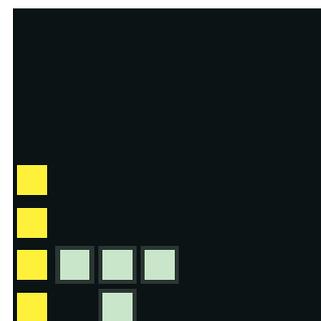


Figure 4 Pengurangan Keuntungan Blok Bawah oleh Blok T

Dari gambar di atas dapat dilihat bahwa keuntungan blok T semula adalah “3” yang didapat dari penetapan nilainya. Akan tetapi ada pengurangan keuntungan akibat 2 buah blok kosong yang diakibatkan oleh blok T. Karena setiap blok kosong yang diakibatkan akan mengurangi keuntungan sebesar “4” maka keuntungan total yang didapat dari susunan di atas adalah  $3 - (4 \times 2) = -5$ .

### 3.2 Greedy Berdasarkan Tingkat

Tingkat yang dimaksudkan di sini adalah tingginya penempatan suatu blok. Tinggi tingkatan ini diukur dari bagian paling bawah blok teracu ke landasan permainan. Hal ini bertujuan agar bagian paling bawah diisi terlebih dahulu sebelum mengisi bagian atas karena semakin tinggi susunan blok tetris maka pemain akan semakin rawan untuk kalah dalam permainan ini.

Untuk melakukan penghitungan keuntungan digunakan prinsip sebagai berikut. Untuk setiap ketinggian yang bertambah “1” maka jumlah keuntungan akibat penempatan blok tersebut akan berkurang “1”. Berikut adalah contoh pengurangan keuntungan akibat tingkat penyusunan blok.

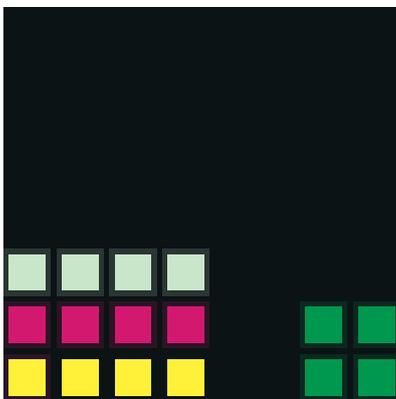


Figure 5 Pengurangan Keuntungan Tingkat oleh Blok I

Dari gambar di atas dapat dilihat bahwa untuk peletakan blok I horizontal berwarna biru akan memberikan keuntungan dari penetapan nilai sebesar “4”. Tetapi dari tingkatannya keuntungan tersebut akan dikurang “2” karena blok tersebut sedang berada di tingkat ke-3. Keuntungan total dari penempatan seperti ini adalah  $4 - 2 = 2$ .

## IV. STUDI KASUS

Berikut adalah berbagai studi kasus dengan memanfaatkan algoritma greedy yang telah dijabarkan di atas.

### 4.1 Studi Kasus I

Kasus I adalah kasus normal dimana blok-blok lain telah tersusun secara acak dan pemain diharapkan dapat menempatkan blok yang akan datang berikutnya ke tempat yang tepat untuk mendapatkan solusi yang optimum.

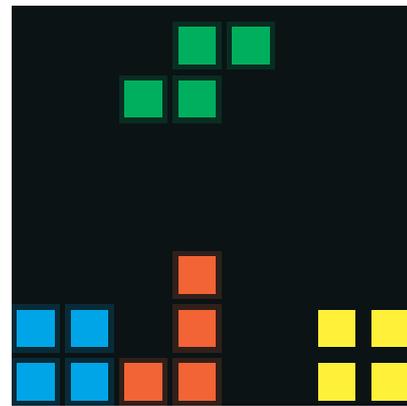


Figure 6 Kondisi Awal Studi Kasus I

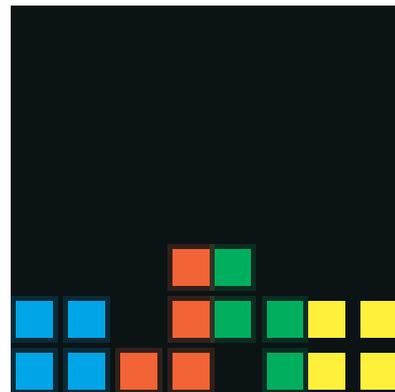


Figure 7 Kondisi Pilihan-1 Studi Kasus I

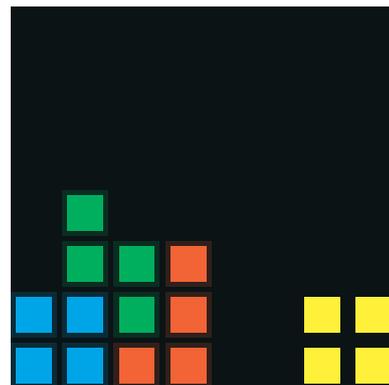


Figure 8 Kondisi Pilihan-2 Studi Kasus I

Dari gambar kondisi awal di atas dapat dilihat bahwa kita akan berusaha menempatkan blok S ke tempat yang paling optimal. Untuk kasus I ini pilihan hanya dibatasi sebanyak dua buah yang digambarkan di dalam gambar kedua dan ketiga.

Untuk setiap pilihan dilakukan penghitungan sebagai berikut :

#### 1) Pilihan 1

Keuntungan dari penetapan nilai = 2

Pengurangan keuntungan akibat blok bawah = -4

Pengurangan keuntungan akibat tingkat = 0

Dari hasil di atas dapat dilihat bahwa total

keuntungan adalah  $2 - 4 - 0 = -2$

## 2) Pilihan 2

Keuntungan dari penetapan nilai = 2  
Pengurangan keuntungan akibat blok bawah = 0  
Pengurangan keuntungan akibat tingkat = -1  
Dari hasil di atas dapat dilihat bahwa total  
keuntungan adalah  $2-0-1 = 1$

Dari analisis kedua pilihan ini maka dapat diputuskan bahwa pilihan 2 adalah pilihan yang lebih menguntungkan daripada pilihan 1. Selain itu dari studi kasus ini juga dapat diketahui bahwa blok bawah yang kosong akan lebih merugikan dibandingkan tingginya tingkat dalam rentang tertentu. Sehingga jika ada blok bawah yang kosong akibat penempatan maka sebaiknya dicari penempatan di tingkat yang lebih tinggi, terkecuali jika tingkat yang akan dipilih sudah terlalu ekstrim sehingga membuat rawan akan kalahnya pemain dalam permainan.

### 4.2 Studi Kasus II

Kasus II adalah kasus pembuktian bahwa algoritma greedy hanya berpatokan terhadap solusi optimum lokal tanpa memperhatikan solusi optimum global dalam mencari solusi di setiap langkah yang dijalaninya. Seperti yang sudah dijelaskan sebelumnya, algoritma greedy hanya berharap jika ia mengambil solusi optimum lokal maka solusi optimum global dapat terpenuhi.

Pengujian kasus II ini cukup sederhana, yaitu dengan dimulai dari awal permainan pun dapat dilihat bahwa algoritma greedy tidak menjamin terpenuhinya solusi optimum global.

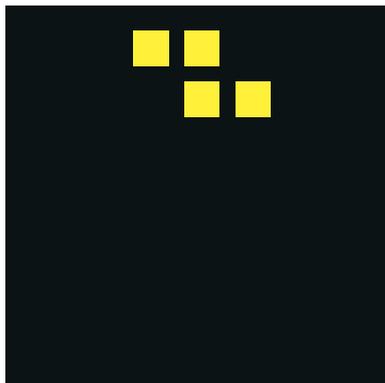


Figure 9 Kondisi Awal Studi Kasus II

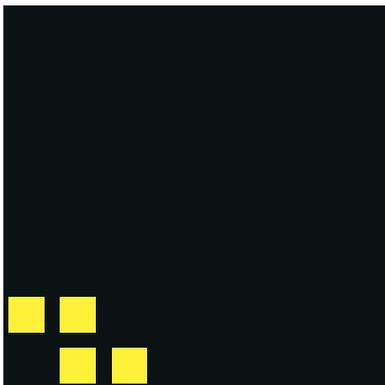


Figure 10 Kondisi Pilihan-1 Studi Kasus II

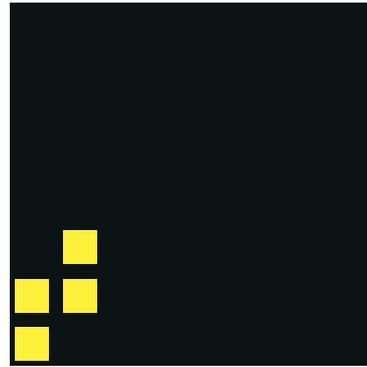


Figure 11 Kondisi Pilihan-2 Studi Kasus II

Dari gambar kondisi awal di atas dapat dilihat bahwa pada awal permainan kita akan menempatkan balok Z ke tempat yang menguntungkan. Berdasarkan algoritma greedy yang telah dijabarkan di atas, berikut adalah penghitungan keuntungan untuk kedua pilihan yang tersedia :

- 1) Pilihan 1  
Keuntungan dari penetapan nilai = 3  
Pengurangan keuntungan akibat blok bawah = -4  
Pengurangan keuntungan akibat tingkat = 0  
Dari hasil di atas dapat dilihat bahwa total  
keuntungan adalah  $3-4-0 = -1$
- 2) Pilihan 2  
Keuntungan dari penetapan nilai = 2  
Pengurangan keuntungan akibat blok bawah = -4  
Pengurangan keuntungan akibat tingkat = 0  
Dari hasil di atas dapat dilihat bahwa total  
keuntungan adalah  $2-4-0 = -2$

Dari hasil penghitungan di atas dapat dilihat bahwa algoritma greedy yang telah didefinisikan akan memilih pilihan 1 sebagai pilihan yang optimal. Padahal jika dipikir dengan logika, pilihan 1 akan berdampak kepada ketidakefektifan penyusunan blok puzzle di tahap-tahap selanjutnya.

Hal ini disebabkan karena pada pilihan 2, bagian kosong tersebut masih dapat diselipkan sehingga menjadi utuh. Hal ini tidak berlaku pada pilihan 1 dimana blok kosong tersebut sudah terisolasi seluruhnya.

Dari hasil analisis ini dapat diketahui bahwa algoritma greedy hanya memperhatikan solusi optimum lokal di setiap langkahnya tanpa memperhatikan apakah langkah tersebut akan menyebabkan solusi optimum global juga di tahap-tahap selanjutnya.

## IV. KESIMPULAN

Kesimpulan yang dapat diperoleh dari hasil pembahasan di makalah ini adalah bahwa penggunaan algoritma greedy berdasarkan blok bawah dan algoritma greedy berdasarkan tingkat dapat digunakan secara bersamaan karena tidak terkait satu sama lain. Hal ini justru akan menambah keefektifan penghitungan keuntungan. Akan tetapi ada syarat tertentu yang perlu

diketahui dalam menggunakan keduanya secara bersamaan. Hal itu adalah bahwa prioritas blok bawah lebih tinggi daripada tingkat dimana nilai blok bawah akan mempengaruhi pengurangan sebesar “4” sementara nilai tingkat hanya mempengaruhi pengurangan sebesar “1” untuk setiap tingkat.

Selain itu dapat diketahui juga bahwa meski menggunakan kedua dasar tersebut secara bersamaan, algoritma greedy tetap tidak dapat menjamin kondisi solusi optimum global terpenuhi. Hal ini dikarenakan algoritma greedy hanya berpatokan terhadap solusi optimum lokal dengan harapan solusi tersebut akan membawanya ke solusi optimum global.

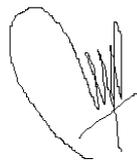
### DAFTAR PUSTAKA

- [1] Munir, Rinaldi. Diktat Kuliah IF3051 Strategi Algoritma. Program Studi Teknik Informatika Institut Teknologi Bandung. 2007
- [2] <http://www.tetris.com/history/index.aspx>  
Waktu akses : 16 Desember 2012, pukul 23.19
- [3] <http://inventors.about.com/od/tstartinventions/a/Tetris.htm>  
Waktu akses : 16 Desember 2012, pukul 23.19
- [4] [www.flash-game-design.com/tutorials/tetris-avatar.html](http://www.flash-game-design.com/tutorials/tetris-avatar.html)  
Waktu akses : 17 Desember 2012, pukul 12.25

### 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, 17 Desember 2012



Mario Orlando Teng  
13510057