

# Strategi Permainan Domino menggunakan Algoritma Greedy

Raihan Maulana Warman / 13514076

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13514076@std.stei.itb.ac.id

**Abstract**—Domino adalah salah satu permainan favorit di kalangan masyarakat Indonesia. Permainannya yang tergolong sederhana tapi mengasyikkan menyebabkan permainan ini menjadi populer. Permainan domino pada umumnya dimainkan oleh 4 orang. Permainan domino ini dapat dengan mudah dimainkan di warung-warung atau di tempat-tempat yang biasa dikunjungi oleh anak muda. Untuk menyelesaikan permainan ini dibutuhkan strategi yang matang dan perhitungan yang tepat. Oleh karena itu digunakan Algoritma Greedy untuk menyediakan solusi yang optimal agar permainan ini dapat dimenangkan.



gambar disadur dari <https://id.wikipedia.org/wiki/Domino>

**Keywords**—Domino, Balak, Gapleh, Kartu,, Optimal, Nilai

## I. PENDAHULUAN

Domino adalah salah satu jenis permainan kartu yang ada di Indonesia. Permainan Domino merupakan permainan yang membutuhkan ketelitian dan pengambilan keputusan yang tepat dari pemain agar permainan ini dapat dimenangkan. Permainan ini dapat dimainkan oleh berbagai orang dari berbagai golongan usia yang beragam.

Permainan domino bisa dikatakan sebagai permainan kartu yang merakyat. Hal ini dikarenakan kemudahan dalam mendapatkan peralatannya. Harga dari peralatannya pun tidak terlalu mahal sehingga permainan ini menjadi salah satu permainan favorit di masyarakat. Dapat dilihat banyak sekali masyarakat yang memainkan permainan domino di warung-warung atau tempat-tempat hiburan lainnya.

Permainan domino ini biasanya dimainkan untuk mengisi waktu luang saja. Permainan domino biasanya membutuhkan waktu kurang lebih 10 menit sampai selesai. Permainan domino biasanya berjalan dengan tempo cepat sehingga permainan domino ini bisa mengakibatkan si pemain menjadi lupa waktu karena keasyikan bermain.

Kartu yang digunakan dalam permainan domino ada yang terbuat dari bahan akrilik dan ada yang terbuat dari bahan kertas. Jumlah kartu pada permainan domino ini berjumlah 28 yang nantinya akan didistribusikan ke semua pemain. Kondisi menang dari permainan ini adalah apabila kartu yang kita punya telah diturunkan semua atau apabila semua pemain tidak bisa menurunkan kartu lagi dan setiap pemain masih memiliki sisa kartu di tangan, maka pemain dengan nominal angkat paling kecil yang menang.

Untuk memenangkan permainan domino ini dibutuhkan berbagai macam strategi yang efektif. Salah satu strategi untuk menyelesaikan permainan domino ini adalah dengan memanfaatkan algoritma *Greedy*. Oleh karena itu, penulis akan membahas pemanfaatan algoritma *greedy* untuk menyelesaikan permainan domino.

## II. DASAR TEORI

### 2.1 Istilah dan peraturan dalam permainan Domino

Berikut adalah istilah yang biasa digunakan dalam permainan domino :

1. Balak  
Adalah kartu yang punya angka sama. Balak paling besar adalah Balak 6. Tapi kalau dalam posisi

permainan berakhir dan kamu masih pegang Balak Kosong tanpa ada kartu dengan angka kosong, dihitungnya jadi 25.

## 2. Gable

Istilah gable dari permainan Gable adalah suatu kondisi permainan yang udah nggak mungkin dilanjutkan lagi karena kartunya udah nggak bisa turun lagi [sudah tidak ada lagi jalan]. Tapi bisa juga salah satu pemain emang sengaja bikin gable, biasanya karena tau dia nggak mungkin habiskan kartunya.

Berikut adalah peraturan dasar yang digunakan saat bermain domino :

1. 4 orang pemain domino
2. Setiap pemain diberi 7 kartu yang telah diacak terlebih dahulu.
3. Bila ada salah satu pemain yang mempunyai minimal 5 balak maka kartu akan diacak kembali.
4. Penyusunan kartu dilakukan secara giliran dan berurutan , biasanya mengikuti arah jarum jam.
5. Bila saat giliran, angka kartu di tangan tidak ada yang cocok dengan angka kartu di meja maka pemain tersebut harus melewati gilirannya.
6. Penyusunan kartu dilakukan sampai ada pemain yang kartunya habis dan permainan harus berhenti bila sudah ada salah satu pemain yang kartunya habis dan langsung di adakan perhitungan angka kartu masing-masing pemain yang kartunya masih tersisa. (yang dihitung jumlah angkanya bukan jumlah kartunya)
7. Pemain yang memiliki jumlah angka kartu yang paling besar itulah yang kalah dan diharuskan mengocok dan membagikan kartu lagi ke para pemain.
8. Pemain yang kartunya habis terlebih dahulu merupakan pemenang.
9. Pemenang pada permainan sebelumnya harus mengeluarkan kartu balak untuk memulai kembali permainan

10. Kartu awal yang dikeluarkan oleh pemain yang menang pada game sebelumnya tidak boleh balak 6 atau balak kosong.

## 2.2 Algoritma Greedy

Algoritma Greedy merupakan salah satu algoritma yang banyak digunakan untuk memecahkan permasalahan optimalisasi. Pemecahan permasalahan optimalisasi bisa dengan dilakukan dengan melakukan maksimalisasi atau dengan minimalisasi. Greedy berasal dari bahasa inggris yang berarti rakus, tamak, atau serakah. Prinsip utama dari algoritma greedy adalah mengambil atau menghasilkan keluaran dengan nilai optimalisasi yang paling baik sesuai dengan prinsip dari algoritma greedy yaitu “ *take what you can get now !!* “.

Algoritma greedy akan mengambil solusi yang paling optimal pada setiap langkahnya. Tapi solusi optimal pada satu langkah bisa jadi tidak mengarah ke solusi paling optimal untuk keseluruhan permasalahan. Beberapa persoalan yang bisa diselesaikan dengan algoritma greedy adalah sebagai berikut :

1. Penukaran uang
2. Persoalan Knapsack
3. Pohon Merentang minimum
4. Lintasan terpendek
5. Traveling Salesman Problem
6. Pemampatan data dengan algoritma Huffman
7. Minimisasi waktu di dalam sistem
8. Penjadwalan job dengan tenggat waktu

### 2.2.1 Skema Umum Algoritma Greedy

Algoritma greedy tersusun oleh elemen-elemen berikut :

1. Himpunan Kandidat  
Himpunan yang berisi elemen-elemen pembentuk solusi dinyatakan dengan simbol C
2. Himpunan Solusi  
Himpunan yang berisi kandidat-kandidat yang terpilih sebagai solusi persoalan dinyatakan dengan simbol S
3. Fungsi Seleksi  
Fungsi yang digunakan untuk memilih solusi lokal yang paling optimal.  
Dinyatakan dengan predikat seleksi
4. Fungsi Kelayakan  
Memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak. Dalam artian

lain, yaitu solusi yang diberikan dapat menyelesaikan permasalahan.  
Dinyatakan dengan predikat layak

5. Fungsi Objektif  
Fungsi Objektif digunakan untuk mengoptimisasi nilai solusi.

Setelah diketahui elemen-elemen algoritma greedy, berikut diberikan langkah-langkah eksekusi algoritma greedy dengan memanfaatkan elemen-elemen tersebut :

- i. Inisialisasi S dengan kosong
- ii. Pilih sebuah kandidat( dengan fungsi Seleksi) dari C
- iii. Kurangi C dengan kandidat yang sudah dipilih dari langkah ii di atas.
- iv. Periksa apakah kandidat yang dipilih merupakan solusi yang layak, jika ya, masukkan kandidat tersebut ke dalam himpunan solusi.
- v. Periksa apakah himpunan solusi sudah memberikan solusi yang lengkap. Jika ya, hentikan program, jika tidak, maka program dilanjutkan.

Langkah-langkah eksekusi algoritma greedy disadur dari Buku Strategi Algoritma(Munir, Rinaldi. 2009).

### 2.2.2 Kekurangan dari Algoritma Greedy

1. Algoritma greedy tidak berhasil untuk beberapa kasus
2. Solusi Lokal Optimum kadang tidak mengarah ke solusi global optimum

### 2.2.3 Kelebihan Algoritma Greedy

1. Penyelesaian masalah lebih cepat karena tidak semua kemungkinan solusi yang diperiksa
2. Pengimplementasian dari algoritma greedy dikatakan lebih mudah dibandingkan dengan algoritma lainnya
3. proses eksekusi dari Algoritma greedy lebih cepat

### 2.3 Pseudocode Algoritma Greedy untuk persoalan domino

*function greedy(input kartu : array of integer, input tepi1, tepi2 : integer) → himpunan\_solusi*

#### DEKLARASI

*n\_kartu : integer // banyak kartu pemain  
loop : integer  
kandidat : himpunan\_kandidat*

*calon : kandidat  
solusi : himpunan\_solusi*

#### ALGORITMA

```

loop ← 1
while ( loop <= n_kartu)
  if (tepi1 = kartu[loop]) or (tepi2 = kartu[loop])
    kartu[loop] U kandidat
    loop ← loop + 1

while (not SOLUSI(S) and kandidat <> {})
  calon ← kandidat // ambil salah satu dari kandidat
  kandidat ← kandidat – calon
  if (LAYAK(S U calon)
    S ← S U calon
  endif
endwhile

if (SOLUSI(S))
  return S
else
  return {}

```

### III. ANALISIS PERMASALAHAN

#### 3.1 Kartu terbaik untuk diturunkan

Untuk kasus kartu terbaik untuk diturunkan,, maka penulis menggunakan dua konsep untuk mencari kartu dengan nilai optimalisasi yang paling baik, yaitu konsep “nilai total terbesar yang akan diturunkan” dan konsep “Jenis kartu yang paling banyak muncul yang akan diturunkan”. Lalu kedua konsep tersebut akan disesuaikan dengan algoritma greedy.

Terdapat beberapa kondisi berbeda untuk menganalisis kartu terbaik untuk diturunkan, kondisi tersebut adalah sebagai berikut :

##### 3.1.1 Kartu yang harus disambung memiliki nilai yang sama



gambar disadur dari [http://2.bp.blogspot.com/-3lpmkzasoyQ/VQZZJycOeSI/AAAAAAAAAII/ji5\\_qZFtpek/s1600/DOMINO%2Bgaple.jpg](http://2.bp.blogspot.com/-3lpmkzasoyQ/VQZZJycOeSI/AAAAAAAAAII/ji5_qZFtpek/s1600/DOMINO%2Bgaple.jpg)

Berdasarkan gambar diatas, maka pemain harus menurunkan kartu yang memiliki angka 3. Oleh karena itu, program akan memeriksa semua kartu yang dimiliki. Apakah ada kartu yang memiliki angka 3 atau tidak.

Misalkan pemain memiliki kartu sebagai berikut di tangan :

Kartu di tangan	Bisa diturunkan
4   1	Tidak
3   2	Ya
5   6	Tidak
4   3	Ya
6   6	Tidak

Lalu algoritma akan memeriksa semua kartu yang dimiliki.

Lalu algoritma akan mencari kartu yang bisa diturunkan terlebih dahulu. Setelah kartu yang bisa diturunkan didapat, algoritma akan memeriksa kartu yang bisa diturunkan tersebut untuk dicari kartu dengan nilai optimalisasi terbaik.

Kartu di tangan yang bisa diturunkan :

Kartu di tangan	Bisa diturunkan
3   2	Ya
4   3	Ya

Setelah didapat kartu yang memenuhi syarat untuk diturunkan, barulah algoritma greedy mencari nilai yang paling optimal. Karena penulis menggunakan dua konsep untuk pengerjaan algoritma greedy, maka kedua konsep tersebut akan dibahas untuk mencari nilai yang paling optimal.

Apabila kita menggunakan konsep “Nilai total terbesar yang akan diturunkan” maka algoritma greedy akan memeriksa nilai total dari tiap-tiap kartu.

Kartu di tangan	Nilai Total
3   2	5
4   3	7

Berdasarkan tabel diatas, maka algoritma greedy akan memilih kartu ( 4 | 3 ) sebagai kartu dengan nilai yang paling optimal.

Apabila digunakan konsep “Jenis kartu yang paling banyak muncul yang akan diturunkan”. Maka program akan mengecek total kartu yang memiliki angka yang berpasangan dengan 3 yang sudah diturunkan.

Kartu di tangan	Total kemunculan
3   2	4
4   3	3

Berdasarkan informasi yang didapat di dalam tabel, total kemunculan dari kartu yang memiliki angka 2 ( 3 | 2 ) yaitu sebanyak 4. Sedangkan total kemunculan kartu yang memiliki angka 4 ( 4 | 3 ) yaitu sebanyak 3. Oleh karena itu algoritma greedy akan memilih kartu ( 3 | 2 ) sebagai kartu dengan nilai yang paling optimal.

Apabila tidak ada kartu yang bisa diturunkan, maka pemain harus melewatkan gilirannya.

### 3.1.2 Kartu yang disambung memiliki nilai berbeda



gambar disadur dari <http://images.malesbanget.com/mbdcposts/2014/01/gaple-560x419.jpg>

Untuk kasus kedua ini, metode algoritma greedy tidak terlalu berbeda dengan kasus pertama. Dari gambar diatas, dapat diketahui bahwa kartu yang diturunkan harus memiliki nilai 5 atau 6. Oleh karena itu, program akan memeriksa terlebih dahulu kartu di tangan pemain yang memiliki nilai 5 atau 6.

Misalkan kartu yang dimiliki pemain adalah sebagai berikut :

Kartu ditangan	Bisa diturunkan
3   5	Ya
2   6	Ya
4   6	Ya
3   3	Tidak
5   6	Ya
3   6	Ya

Setelah didapat kartu yang bisa diturunkan , maka algoritma greedy akan mencari nilai yang paling optimal berdasarkan konsep “Jenis kartu yang paling banyak muncul

yang akan diturunkan” dan konsep “Nilai total terbesar yang akan diturunkan”.

Apabila digunakan konsep “Nilai total terbesar yang akan diturunkan”, maka program akan mencari nilai total dari kartu yang dimiliki pemain.

Kartu di tangan	Nilai Total
3   5	8
2   6	8
4   6	10
5   6	11
3   6	9

Berdasarkan informasi yang didapat pada tabel diatas, maka kartu ( 3 | 5 ) memiliki nilai total 8, kartu ( 2 | 6 ) memiliki nilai total 8, kartu ( 4 | 6 ) memiliki nilai total 10, kartu ( 5 | 6 ) memiliki nilai total 11, kartu ( 3 | 6 ) memiliki nilai total 9.

Berdasarkan data yang telah didapat, maka kartu dengan nilai yang paling optimal adalah kartu ( 5 | 6 ). Oleh karena itu, algoritma greedy akan memilih kartu ( 5 | 6 ) untuk diturunkan.

Apabila konsep “Jenis kartu yang paling banyak muncul yang akan diturunkan” digunakan. Maka program akan mengecek total kartu yang memiliki angka yang berpasangan dengan 5 atau 6 yang sudah diturunkan.

Kartu di tangan	Total Kemunculan
3   5	2
2   6	5
4   6	5
5   6	3 (kemunculan 5 ) atau 2 (kemunculan 6)
3   6	2

Berdasarkan informasi yang didapat di dalam tabel, total kemunculan dari kartu yang memiliki angka 3 ( 3 | 5 ) dan ( 3 | 6 ) yaitu sebanyak 2. Total kemunculan kartu yang memiliki angka 2 ( 2 | 6 ) yaitu sebanyak 5. Total kemunculan angka 4 ( 4 | 6 ) yaitu sebanyak 5 dan total kemunculan angka 5 ( 5 | 6 ) yaitu sebanyak 3 dan total kemunculan angka 6 ( 5 | 6 ) yaitu sebanyak 2. Oleh karena itu algoritma greedy akan memilih kartu ( 2 | 6 ) dan kartu ( 4 | 6 ) sebagai kartu dengan nilai yang paling optimal.

Karena algoritma memiliki dua solusi untuk persoalan ini, maka akan digunakan konsep “Nilai total terbesar yang akan diturunkan” untuk mencari kartu yang memiliki nilai paling optimal diantara dua solusi yang tersedia.

Kartu di tangan	Nilai Total
2   6	8
4   6	10

Karena nilai total kartu ( 4 | 6 ) lebih besar daripada nilai total kartu ( 2 | 6 ), maka kartu yang dipilih sebagai solusi terbaik adalah kartu ( 4 | 6 ).

Apabila pemain tidak memiliki satupun kartu yang bisa diturunkan, maka pemain tersebut harus melewatkan gilirannya.

#### IV. KESIMPULAN

Berdasarkan Analisis permasalahan yang telah dibahas sebelumnya, maka didapat beberapa kesimpulan sebagai berikut :

1. Algoritma Greedy dapat diimplementasikan untuk mencari solusi dalam permainan domino
2. Solusi yang dihasilkan algoritma greedy bisa jadi merupakan solusi optimal atau bisa jadi merupakan solusi yang tidak optimal
3. Untuk mendapatkan algoritma yang lebih efisien dari pada algoritma greedy untuk penanganan persoalan domino, akan dibutuhkan Kecerdasan Buatan ( AI ) yang sangat baik
4. Permainan domino bukan termasuk judi

#### V. REFERENSI

[1] Munir, Rinaldi. 2009. Strategi Algoritma. Bandung :Penerbit Institut Teknologi Bandung.

[2] Torikiri. 2013. Algoritma Greedy. <https://torikiri.wordpress.com/2013/09/10/algoritma-greedy-2/> diakses pada tanggal 8 Mei 2016.

[3] Pohan. 2015. Aturan, Tips, dan Trik Main Gapple . <http://linikini.id/news/236/> diakses pada tanggal 8 Mei 2016.

[4] Achmad, Hari. 2012 . Pengertian dan Tips bermain Gapple . <http://yourdayandtechno.blogspot.co.id/2013/08/pengertian-peraturan-dan-tips.html>. Diakses pada tanggal 8 Mei 2016.

[5] Jufriyanto, Muhammad . 2009. Algoritma Greedy. <https://bertzzie.com/knowledge/analisis-algoritma/Greedy.html> diakses pada tanggal 7 Mei 2016.

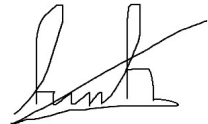
[6] Efed, Ilham . 2015 . Pengertian Algoritma Greedy . <http://www.it-jurnal.com/2015/09/pengertian-algoritma-greedy.html>. Diakses pada tanggal 8 Mei 2016.

- [7] Unknown . Data Structures - Greedy Algorithms . [Http://tutorialpoints.com/data\\_structures\\_algorithms/greedy\\_algorithms.html](http://tutorialpoints.com/data_structures_algorithms/greedy_algorithms.html). Diakses pada tanggal 7 Mei 2016
- [8] Boruvka . Greedy Algorithm | Set 9 ( Boruvka's Algorithm ) . <http://www.geeksforgeeks.org/greedy-algorithms-set-9-boruvkas-algorithm/>. Diakses pada tanggal 8 Mei 2016

## 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, 8 Mei 2016



Raihan Maulana Warman  
13514076