

# Penerapan Algoritma Greedy pada Permainan 2048

Afrizal Sebastian - 13520120

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail (gmail): 13520120@std.stei.itb.ac.id

**Abstract**—Permainan 2048 merupakan suatu permainan bejenis puzzle yang dimainkan dengan cara mengeser kotak yang memiliki nilai didalamnya. Permainan ini memiliki ukuran yang pada umumnya 4x4 tile dan dimainkan secara perseorangan. Nilai yang ada pada kotak/tile adalah perpangkatan 2 dan goal dari permainan ini adalah membuat kotal/tile dengan nilai setinggi mungkin. Pada makalah ini akan dibahas penerapan algoritma greedy pada permainan 2048 untuk mendapatkan nilai tile/kotak yang tinggi.

**Keywords**—Permainan 2048, Algoritma Greedy, Puzzle, Kotak/Tile.

## I. PENDAHULUAN

Pada saat ini telah banyak algoritma yang ditemukan untuk menyelesaikan persoalan-persoalan yang ada, Algoritma Greedy merupakan salah satu algoritma yang pada umumnya digunakan untuk mencari nilai optimal dari suatu persoalan (optimasi). Algoritma ini masih mampu menyelesaikan banyak persoalan optimasi dan masih banyak digunakan sampai saat ini.

Salah satu penggunaan algoritma ini adalah dalam permainan 2048. Permainan 2048 merupakan salah satu permainan optimasi, yaitu membentuk suatu kotak/tile dengan nilai 2048. Namun, pada umumnya permainan ini tidak menghentikan permainan hingga nilai 2048, permainan dapat dilanjut hingga kotak/tile memiliki nilai setinggi mungkin hingga puzzle tidak dapat digerakan lagi.

Permainan 2048 ini diciptakan oleh pengembang web yang berasal dari itali yaitu Gabriele Cirulli pada Maret 2014. Permainan ini dilakukan oleh perseorangan dan dapat dimainkan dengan mudah, hanya melakukan penggeseran terhadap kotak dan menyatukan kotak-kotak dengan nilai yang sama.



Gambar 1.1 Gameplay Permainan 2048

Diakses pada 20 Mei 2022, 17:20 WIB

Sumber : <https://play2048.co/>

Pada Makalah ini akan dibahas penerapan Algoritma Greedy dalam memainkan permainan 2048. Algoritma greedy akan digunakan untuk mendapatkan nilai 2048 atau nilai setinggi tingginya

## II. DASAR TEORI

### A. Definisi Algoritma

Algoritma adalah metode atau langkah yang direncanakan tersusun dan berurutan untuk menyelesaikan atau memecahkan suatu permasalahan dengan sebuah instruksi atau kegiatan.

Pada dasarnya, fungsi utama algoritma adalah untuk memecahkan suatu permasalahan. Algoritma pada umumnya digunakan untuk membantu setiap orang yang ingin menkonversikan sebuah permasalahan ke suatu bahasa permograman.

Algoritma berbeda dengan program. Algoritma merupakan sebuah desain sebelum membangun mesin dalam rekayasa perangkat lunak. Sedangkan program adalah hasil implementasi dari desain yang telah dibuat dalam bentuk baris-baris code.

Menurut Donald E.Knuth, algoritma memiliki 5 kriteria, yaitu Finiteness, Definiteness, Input, Output, dan Effectiveness.

### B. Algoritma Greedy

#### 1. DEFINISI

Algoritma Greedy adalah algoritma yang memecahkan persoalan secara langkah per langkah sedemikian sehingga, pada setiap langkah :

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

Algoritma greedy merupakan salah satu metode yang paling populer dan sederhana dalam memecahkan permasalahan optimasi.

## 2. ELEMEN-ELEMEN ALGORITMA GREEDY

Elemen-elemen algoritma greedy adalah:

- Himpunan Kandidat,  $C$  : berisi kandidat yang akan dipilih pada setiap langkah.
- Himpunan Solusi,  $S$  : berisi kandidat yang sudah dipilih.
- Fungsi Solusi : menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi.
- Fungsi Seleksi (*selection function*) : memilih kandidat berdasarkan strategi greedy tertentu. Strategi greedy ini bersifat heuristic.
- Fungsi Kelayakan (*feasible*) : memeriksa apakah kandidat yang dipilih dapat dimasukkan ke dalam himpunan solusi.
- Fungsi Obyektif : memaksimumkan atau meminimumkan.

Maka dapat dikatakan bahwa, Algoritma greedy melibatkan pencarian sebuah himpunan bagian,  $S$ , dari himpunan kandidat  $C$ ; yang dalam hal ini,  $S$  harus memenuhi kriteria yang ditentukan, yaitu  $S$  menyatakan suatu solusi dan  $S$  dioptimasi oleh fungsi obyektif.

Skema Umum Algoritma Greedy :

```

function greedy(C : himpunan_kandidat) → himpunan_solusi
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy }
Deklarasi
x : kandidat
S : himpunan_solusi
Algoritma:
  S ← {} { inisialisasi S dengan kosong }
  while (not SOLUSI(S) and (C ≠ {})) do
    x ← SELEKSI(C) { pilih sebuah kandidat dari C }
    C ← C - {x} { buang x dari C karena sudah dipilih }
    if LAYAK(S ∪ {x}) then { x memenuhi kelayakan
    untuk dimasukkan ke dalam himpunan solusi }
      S ← S ∪ {x} { masukkan x ke dalam
    himpunan solusi }
  endif
endwhile
{ SOLUSI(S) or C = {} }
if SOLUSI(S) then { solusi sudah lengkap }
  return S
else
  write('tidak ada solusi')
endif
    
```

## 3. CONTOH PERSOALAN

Contoh persoalan yang dapat diselesaikan oleh algoritma greedy adalah:

- Persoalan penukaran uang.
- Persoalan memilih aktivitas.
- Minimasi waktu di dalam sistem.
- Persoalan *knapsack*
- Persoalan *job* dengan tenggat waktu.
- Pohon merentang minimum.
- Lintasan terpendek.
- Kode Huffman.
- Pecahan Mesir.

## C. Game

Pengertian game pada masa teknologi adalah, permainan yang deprogram pada suatu perangkat yang dapat dijalankan secara *offline* dan *online*.

Ada beberapa manfaat dari game, yaitu *game* dapat sebagai media hiburan, *game* dapat membantu mengasah otak, *game* dapat melatih emosi, sarana mengembangkan diri, dan menyeimbangkan koordinasi tubuh.

Ada beberapa jenis game berdasarkan genrenya, yaitu *game action, fighting, racing, Sports, Role Playing Game, Adventure. Strategy*, dan *puzzle*.

## D. Permainan 2048

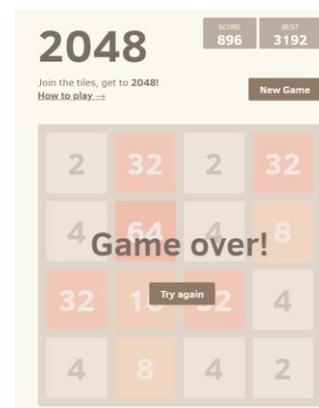
### 1. GOAL PERMAIANAN

Goal/Tujuan dari permainan ini adalah untuk mendapatkan kotak/tile yang memiliki nilai 2048. Namun pada kenyataannya, permainan ini tidak berhenti hanya untuk mendapatkan nilai 2048 pada suatu kotak/tile, namun untuk mendapatkan nilai setinggi-tingginya pada suatu kotak/tile. Permainan akan berakhir jika kotak/tile tidak dapat digerakkan lagi.



Gambar 2.1 Keadaan Menang permainan 2048  
Diakses pada 21 Mei 2022, 08:38 WIB.

Sumber :  
[https://upload.wikimedia.org/wikipedia/commons/f/f9/2048\\_winn.png](https://upload.wikimedia.org/wikipedia/commons/f/f9/2048_winn.png)



Gambar 2.2 Keadaan Kalah permainan 2048

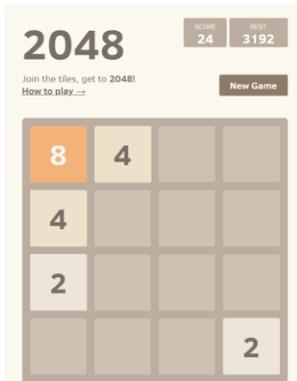
Diakses pada 21 Mei 2022, 08:41 WIB.

Sumber : <https://play2048.co/>

## 2. ATURAN PERMAINAN

Permainan ini telah banyak dikembangkan di berbagai platform seperti *Web*, *Android*, dan *iOS*. Permainan ini dimainkan oleh perseorangan (*Single Player*). Adapun aturan dan cara bermain dari Permainan 2048 ini adalah :

- Puzzle dapat di-*swipe* (digeser) ke 4 arah yaitu atas, kanan, bawah, dan kiri. Setiap melakukan *swipe*, sisi seluruh kotak/tile akan “merapat” ke arah yang di-*swipe*.



Gambar 2.2 Contoh Puzzle sebelum di-*swipe*  
Diakses pada 21 Mei 2022, 10:05 WIB.

Sumber : <https://play2048.co/>



Gambar 2.3 Contoh Puzzle setelah melakukan *swipe* kanan.

Diakses pada 21 Mei 2022, 10:07 WIB.

Sumber : <https://play2048.co/>

- Kotak/tile yang memiliki nilai yang sama dan bersebelahan pada saat di *swipe* akan menyatu dan membentuk tile baru dengan nilai hasil penjumlahan 2 tile sebelumnya.



Gambar 2.4 Contoh Puzzle sebelum menyatu  
Diakses pada 21 Mei 2022, 10:09 WIB.

Sumber : <https://play2048.co/>



Gambar 2.5 Contoh Puzzle Setelah menyatu  
Diakses pada 21 Mei 2022, 10:11 WIB.

Sumber : <https://play2048.co/>

- Kotak/Tile baru akan terbentuk pada sisi yang berlawanan dengan arah *swipe* secara random dan nilai yang random pula. Contoh : Jika melakukan *swipe* ke arah kanan, maka akan terbentuk kotak/tile baru pada sisi kiri puzzle.
- Permainan berakhir jika tidak puzzle tidak dapat digerakkan kearah manapun.

## III. PENERAPAN

### A. Asumsi dan Batasan

Penerapan Algoritma *greedy* pada makalah ini, ada beberapa asumsi dan batasan untuk mempersempit area pembahasan, seperti :

1. Kotak/tile yang paling tinggi nilainya diusahakan di letakkan pada sudut kanan bawah.
2. Dari pernyataan (1), maka gerakan yang dominan pada pembahasan ini adalah gerakan ke bawah dan ke kanan, sedangkan dua gerakan lainnya dilakukan jika gerakan utama tidak dapat dipakai.

### B. Penerapan Algoritma Greedy

Dengan menggunakan prinsip *greedy*, maka penerapan algoritma *greedy* pada permainan ini adalah mengambil dua kotak/tile yang bersebelahan dengan nilai tertinggi pada

Puzzle, lalu melakukan gerakan dominan (ke kanan atau ke bawah) untuk menyatukan kotak/tile. Hal ini Dilakukan pada setiap langkah.

Dengan penerapan Algoritma Greedy diatas, maka permainan 2048 dapat dipetakan menjadi elemen-elemen pada algoritma greedy, yaitu :

1. Himpunan Kandidat pada kasus ini adalah semua gerakan yang mungkin pada setiap tahapnya, yaitu gerakan ke atas, ke bawah, ke kanan, atau ke kiri.

2. Himpunan Solusi pada kasus ini adalah gerakan yang diambil pada setiap langkahnya mengikuti algoritma greedy diatas. Gerakan pada setiap langkahnya hanya dapat satu gerakan.

3. Fungsi Solusi pada kasus ini adalah fungsi yang digunakan untuk memilih arah gerakan yang akan diambil pada setiap langkahnya.

4. Fungsi Seleksi pada kasus ini adalah fungsi yang mencari 2 nilai kotak/tile yang berdekatan yang memiliki nilai tertinggi pada puzzle.

5. Fungsi Kelayakan pada kasus ini adalah fungsi yang digunakan untuk mengetahui kelayakan gerakan pada saat dilakukan.

6. Fungsi Obyektif pada kasus ini adalah gerakan pada setiap tahap merupakan gerakan yang memberikan kotak/tile tertinggi pada tahap tersebut.

### C. Implementasi

Dari penjabaran elemen-elemen maka implementasi algoritma greedy tersebut dalam bentuk *pseudocode* adalah sebagai berikut :

```

procedure greedyGame2048 (input/ouput G :
Game2048, output S : Step)
{ I.S : G berisikan papan permainan dengan
kondisi awal, yaitu terdapat 2 kotak/tile
yang tersebar di Puzzle}
{ F.S : G sampai pada kondisi akhir, yaitu
G tidak dapat digerakkan lagi}

Kamus Lokal
    tile1, til2 : integer

Algoritma
    S <-- {}
    while (not (gameOver(G)) do
        tile1, tile2 <-- findMax2Tile(G)
        if (availableSwipe("Kanan", tile1,
tile2) or availableSwipe("Bawah", tile1,
tile2) then {gerakan diambil berdasarkan

```

```

prioritas}
        if (avaiableSwipe("Kanan",
tile1, tile2) then
            swipePuzzle(G, "Kanan")
            S <-- S ∪ {"Kanan"}
        else
            swipePuzzle(G, "Bawah",
tile1, tile2)
            S <-- S ∪ {"Bawah"}
        endif
        else if (avaiableSwipe("Kiri",
tile1, tile2) then
            swipePuzzle(G, "Kiri")
            S <-- S ∪ {"Kiri"}
        else
            swipePuzzle(G, "Atas", tile1,
tile2)
            S <-- S ∪ {"Atas"}
        endif
    endwhile

```

Fungsi dan Procedure tambahan

```

function gameOver(input G : game2048) -->
Boolean
{Mengembalikan true jika permainan 2048
tidak dapat digerakkan lagi}

function findMax2Tile(input G : game2048 )
--> integer, integer
{Mengembalikan index dari 2 tile dengan
nilai tertinggi dan bersebelahan}

function avaiableSwipe(input G : game2048,
input gerakan : String, input tile1 :
integer, input tile2 : integer) -->
Boolean
{Mengembalikan true jika gerakan pada tile
bersangkutan dapat dilakukan}

procedure swipePuzzle (input/output G :
game2048, input gerakan : string)
{I.S G memiliki 2 tile yang berdekatan
dengan nilai yang sama}

```

{F.S tile yang bersebalah telah menjadi satu tile pada G}

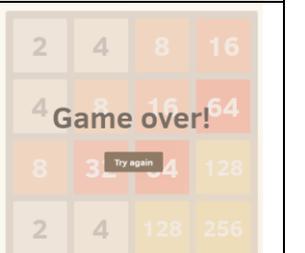
#### IV. PENGUJIAN DAN PEMBAHASAN

##### A. Pengujian

Algoritma Greedy pada Bab III akan dilakukan secara manual karena tidak ada program yang dibuat. Hal tersebut dikarenakan, pembuatan Game 2048 yang cukup rumit serta implementasi algoritma greedy yang mungkin akan cukup berbeda pada saat dibuat dalam bentuk source code.

Oleh karena itu, Pengujian algoritma pada Bab III akan dilakukan secara manual sebanyak 5 kali dengan hasil sebagai berikut:

Tabel 4.1 Hasil Percobaan Algoritma Greedy pada Permainan 2048

Percobaan Ke -	Nilai Kotak/Tile terbesar pada Puzzle	Skor Akhir	Keadaan Akhir
1	128	1772	
2	256	4124	
3	256	4152	

4	128	1820	
5	128	1600	

Dari lima percobaan diatas, didapatkan kotak/tile tertinggi yang terbentuk adalah 256 dengan Skor akhir yaitu 4152. Hasil tersebut didapat pada percobaan ke-3.

##### B. Pembahasan

Dari Percobaan yang telah dilakukan pada Subbab diatas, algoritma greedy yang dipakai pada makalah ini hanya dapat memperoleh Kotak/tile tertinggi sebesar 256 dengan total score 4152.

Dari percobaan diatas juga dapat dilihat bahwa algoritma greedy berupaya membuat kotak/tile yang bernilai paling tinggi berada di sudut kanan bawah. Namun terdapat percobaan yang gagal membuat kotak/tile yang terbesar berada di bawah kanan, yaitu pada percobaan ke-5.

Pada Percobaan ke-5, Algoritma ini gagal membuat kotak/tile tertinggi berada di kanan bawah. Hal ini dapat terjadi karena letak kotak/tile yang baru pada saat *swipe* terjadi dilakukan secara random oleh program. Oleh karena kotak/tile terbesar tidak berada di kanan-bawah *puzzle*, percobaan ke-5 ini merupakan percobaan dengan skor terendah dari semua percobaan, yaitu sebesar 1600 saja.

Dari hasil percobaan diatas juga, membuktikan bahwa algoritma greedy yang digunakan pada makalah ini hanya mampu untuk mendapatkan kotak/tile dengan nilai antara 128 dan 256.

#### V. KESIMPULAN

Algoritma greedy merupakan salah satu algoritma yang dapat digunakan pada permainan 2048. Algoritma ini mengambil keputusan yang paling menguntungkan pada setiap langkah yang diambil pada saat bermain. Dari beberapa percobaan diatas, algoritma ini dapat dijalankan dengan baik.

Namun, algoritma yang dipakai pada makalah ini tidak dapat mencapai goal dari permainan yaitu mendapat angka 2048 pada sebuah kotak/tile. Algoritma yang dipakai hanya mendapatkan kotak/tile dengan nilai terbesar 256 dan dengan skor akhir 4152.

Algoritma ini tidak dapat secara maksimal memainkan permainan 2048 ini karena kotak/tile yang baru muncul sesaat setelah *swipe* terletak secara random, sehingga algoritma ini tidak dapat meng-*handle* gerakan yang mungkin akan dilakukan kedepannya. Permainan 2048 ini mungkin akan lebih efektif dilakukan dengan algoritma *branch and bound* atau *Dynamic Programming*.

#### VIDEO LINK AT YOUTUBE

Berikut adalah lampiran *link* presentasi hasil makalah saya:

<https://youtu.be/eZwHCxFz3aw>

#### UCAPAN TERIMA KASIH

Puji syukur penulis kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya makalah “Penerapan Algoritma Greedy pada Permainan 2048” dapat selesai dengan baik dan tepat waktu.

Penulis mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T sebagai dosen pengajar Mata Kuliah Matematika Diskrit Kelas 03 atas bimbingan dan pengajaran terkait kuliah ini. Penulis juga mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T. yang telah menyediakan materi kuliah Matematika Diskrit, video pembelajaran, serta website yang membantu pengerjaan makalah ini. Penulis juga mengucapkan terima kasih kepada

orangtua, keluarga, teman, serta semua orang yang membantu dalam pengerjaan makalah ini

#### REFERENSI

- [1] <https://play2048.co/> Diakses pada 20 Mei 2022 Pukul 17:20 WIB.
- [2] <https://www.unida.ac.id/teknologi/artikel/pengertian-algoritma.html> Diakses pada 21 Mei 2022 Pukul 08:10 WIB.
- [3] <https://www.ekrut.com/media/algoritma-adalah> Diakses pada 21 Mei 2022 Pukul 08:10 WIB.
- [4] <https://www.nesabamedia.com/pengertian-game/> Diakses pada 21 Mei Pukul 08:30 WIB.
- [5] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf) Diakses pada 21 Mei Pukul 08:15 WIB
- [6] [https://id.wikipedia.org/wiki/2048\\_\(permainan\)](https://id.wikipedia.org/wiki/2048_(permainan)) Diakses pada 21 Mei 08:45 WIB

#### 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, 20 Mei 2022



Afrizal Sebastian 13520120