

Perbandingan Algoritma Brute Force dengan dan tanpa Metode Heuristik pada Permainan Battleship

Ananda Yulizar Muhammad 13518088
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13518088@std.stei.itb.ac.id

Abstract—Battleship merupakan sebuah game strategi klasik yang dilakukan oleh 2 orang pemain. Tujuan dari permainan ini adalah untuk menebak posisi dari semua kapal musuh, dan menenggelamkannya. Kunci dari permainan ini adalah menerapkan strategi yang baik untuk menenggelamkan semua kapal musuh sebelum semua kapal kita ditenggelamkan. Dalam makalah ini, penulis akan menggunakan algoritma *Brute Force* untuk menyelesaikan permainan ini. Pada makalah ini juga akan diperlihatkan perbedaan algoritma dengan dan tanpa menggunakan heuristik.

Keywords—*Brute Force; Battleship*

I. PENDAHULUAN

Algoritma *brute force* adalah sebuah algoritma yang paling mudah untuk diterapkan pada setiap masalah dibanding dengan algoritma-algoritma lain. Namun, untuk membayar kemudahan dalam menerapkan algoritma tersebut, *brute force* cenderung memiliki kompleksitas waktu yang lama dikarenakan ia harus mengiterasi semua kemungkinan yang dapat terjadi hingga ia menemukan solusi yang diharapkan. Algoritma *Brute Force* juga dapat menjadi dasar algoritma penyelesaian sebuah masalah sebelum mengembangkannya menjadi algoritma lain.

Algoritma-algoritma, salah satunya adalah algoritma *brute force*, dapat diterapkan pada permainan-permainan yang membutuhkan strategi. Salah satu permainan tersebut adalah sebuah permainan dari zaman Perang Dunia I yang diciptakan pada tahun 1931, yaitu Battleship. Battleship merupakan permainan strategi yang bertujuan untuk menebak lokasi dari kapal musuh yang tersembunyi, dan menenggelamkannya. Permainan ini dapat dimainkan oleh berbagai kalangan, baik anak-anak, remaja, ataupun orangtua dengan berbagai strategi yang digunakan masing-masing. Pada awal penciptaan, battleship dimainkan dengan menggunakan kertas dan pensil, namun pada tahun 1967, permainan ini dirilis dalam versi



Gambar 1: Cover Permainan BattleShip
(Sumber: <https://board-games-galore.fandom.com/wiki/Battleship>)

permainan papan oleh Milton Bradley.

II. LANDASAN TEORI

A. Algoritma *Brute Force*

Algoritma *brute force* merupakan suatu algoritma pemecahan masalah dengan pendekatan yang lempang. Algoritma *brute force* menggunakan pendekatan yang didasarkan pada pernyataan masalah dan juga definisi konsep yang dilibatkan. Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung, dan jelas. Slogan dari algoritma *brute force* adalah “*Just do it!*”. Algoritma *brute force* juga sering disebut dengan algoritma naif.

Terdapat beberapa karakteristik yang menggambarkan algoritma *brute force*. Karakteristik dari algoritma *brute force* adalah sebagai berikut:

1. Algoritma *brute force* pada umumnya tidak cerdas dan tidak mangkus dikarenakan jumlah komputasi yang harus dilakukan cukup banyak karena ia mengiterasi semua kemungkinan jawaban yang menyebabkan kompleksitas waktu yang lama.
2. Algoritma *brute force* lebih cocok untuk penyelesaian yang kecil. Hal ini dikarenakan untuk penyelesaian yang kecil, kompleksitas waktu antara algoritma *brute force* dengan algoritma lainnya tidak terlalu berbeda.
3. Algoritma *brute force* sering digunakan sebagai basis pembanding dengan algoritma lain yang lebih cerdas.
4. Hampir semua persoalan dapat diselesaikan dengan algoritma *brute force* walaupun penyelesaiannya bukan penyelesaian yang paling mangkus.

Kelebihan dari algoritma *brute force* adalah sebagai berikut:

1. Dapat digunakan untuk menyelesaikan sebagian besar masalah.
2. Sederhana dan mudah dimengerti.
3. Algoritma yang dihasilkan layak untuk masalah-masalah umum.

Algoritma yang dihasilkan merupakan algoritma Kelemahan dari algoritma *brute force* adalah sebagai berikut:

1. Jarang menghasilkan algoritma yang mangkus.
2. Algoritmanya cenderung lambat
3. Tidak sekonstruktif/sekreatif algoritma pemecahan masalah lainnya.

Dalam perancangan algoritma *brute force*, untuk mengurangi jumlah iterasi yang dilakukan dan mempercepat waktu penyelesaian masalah, dapat digunakan teknik heuristik. Heuristik adalah teknik yang dirancang untuk memecahkan persoalan dengan mengabaikan apakah solusi tersebut dapat terbukti secara matematis. Teknik heuristik berbasiskan pengalaman, dari proses pembelejaraan, dan penemuan solusi meskipun tidak dijamin optimal. Walaupun tidak dapat dibuktikan secara matematis, namun dapat sangat berpengaruh terhadap penerapan algoritma *brute force*, terutama dalam penyelesaian masalah dengan skala yang besar.

B. Permainan Battleship

Battleship merupakan sebuah permainan strategi dari zaman Perang Dunia I. Pertama kali dirilis pada tahun 1931 dengan nama *Salvo* di Amerika Serikat oleh Perusahaan Starex. Permainan ini dimainkan dengan menggunakan pensil dan kertas yang telah dicetak dengan arena permainannya. Pada tahun 1967, Milton Bradley merilis versi permainan papan dari battleship dengan menggunakan papan, miniatur kapak plastik, dan pasak berwarna merah dan putih (tergantung versi) untuk diletakkan pada papan tersebut. Tujuan dari permainan ini adalah menenggelamkan semua kapal lawan sebelum lawan menenggelamkan semua kapal kita.

Untuk memulai permainan, kedua pemain menyiapkan masing-masing papan permainannya, atau jika tidak ada dapat menggunakan kertas. Masing-masing player mempunya dua arena, yaitu arena atas dan arena bawah. Tiap arena berisikan kotak-kotak berukuran 10 x 10.



Arena bawah merupakan tempat kita meletakkan kapal kita.

Gambar 2: Papan Permainan BattleShip

(Sumber:

<https://www.videoamusement.com/carnival-games-rental/giant-battleship-game/>)

Tiap kapal memiliki ukuran masing-masing. Ukuran dari kapal berupa berapa banyak kotak yang ditempati oleh kapal tersebut dalam satu garis. Kapal dapat diletakkan dalam posisi vertikal maupun horizontal, namun tidak bisa diagonal. Tidak boleh ada kapal yang menimpa kapal lain. Musuh tidak boleh melihat letak dari kapal-kapal kita. Dalam permainan ini, terdapat beberapa versi mengenai jenis dan jumlah kapal yang dimiliki

oleh seorang pemain. Oleh karena itu, dalam makalah ini versi yang digunakan adalah versi Milton Bradley tahun 1990 sebagai berikut.

No.	Jenis	Ukuran
1	Aircraft Carrier	5
2	Battleship	4
3	Cruiser	3
4	Submarine	3
5	Destroyer	2

Tabel 1. Daftar kapal yang dimiliki pemain

Berikut merupakan contoh gambaran dari peletakkan kapal pemain. Kotak berwarna merah merupakan representasi dari kapal yang diletakkan.

	0	1	2	3	4	5	6	7	8	9
0										
1			█	█	█	█	█			
2										
3	█	█	█				█	█		
4										
5										
6						█	█	█	█	
7		█								
8		█								
9		█								

Tabel 2. Gambaran arena bawah.

Arena atas merupakan gambaran dari peletakkan kapal lawan, akan tetapi tidak diperlihatkan di mana lawan meletakkan kapal mereka. Kita akan menebak posisi dari kapal lawan dengan menandai kotak pada arena atas kita.

Permainan dilakukan secara bergiliran. Masing-masing pemain menebak lokasi dari kapal pemain lawan dengan menggunakan 'misil'. Pemain menyebutkan sebuah koordinat pada arena, dan apabila koordinat tersebut sama dengan koordinat lawan meletakkan kapal mereka, maka lawan harus mengatakan "hit" atau sejenisnya yang menandakan bahwa tebakan kita benar. Apabila tebakan kita salah, maka lawan akan mengatakan "miss" atau sejenisnya. Pada arena atas kita, kotak yang "hit" ditandai dengan tanda O pada kertas atau pasak merah pada papan, dan kotak yang "miss" ditandai dengan tanda X pada kertas atau pasak putih pada papan. Hal ini dilakukan untuk memperoleh gambaran dari formasi kapal lawan. Hal yang sama juga kita lakukan pada formasi kapal kita untuk mengetahui status dari formasi kita. Apabila kotak yang dinaungi sebuah kapal telah "hit" semua, maka kapal tersebut telah tenggelam dan lawan harus memberitahunya kepada kita termasuk dengan jenisnya. Apabila kapal lawan

telah tenggelam semua, maka kita memenangkan permainan, dan sebaliknya.

	0	1	2	3	4	5	6	7	8	9
0	X	X								
1			O	O	O					
2										
3	O	O	O		X	X				
4										
5										
6				X	X					
7										
8										
9										

Tabel 3. Gambaran arena atas

III. PEMBAHASAN

Pada bagian ini akan dijelaskan penerapan algoritma *brute force* pada permainan battleship. Penerapan algoritma *brute force* pada permainan battleship cukup sederhana, yaitu mengiterasi seluruh kotak pada arena lawan, dan apabila menemukan kotak di mana kita mendapat "hit", maka pencarian akan difokuskan ke sekitar kotak "hit" tersebut untuk mendapatkan kapal lawan. Dalam perancangan algoritma *brute force* akan ditambahkan beberapa fungsi heuristik untuk membandingkan kompleksitas waktu sebelum dan sesudah diberikan fungsi heuristik.

Penulis membuat sebuah pengujian sederhana menggunakan bahasa pemrograman Python. Pada pengujian ini, kita hanya akan menebak formasi kapal lawan saja sedangkan kita tidak akan membuat formasi kapal dan tidak ada lawan yang akan menyerang kita. Oleh karena itu, penerapan strategi ini tidak menjamin kemenangan dalam permainan ini.

A. Penerapan Algoritma *Brute Force* pada Battleship

Dalam penggunaan algoritma *brute force* pada permainan battleship, strategi yang digunakan cukup sederhana. Berikut merupakan langkah-langkah strategi *brute force* dalam permainan battleship.

1. Membuat array kapal yang dimiliki oleh lawan
2. Jika array kapal yang dimiliki lawan telah kosong, maka kita telah memenangkan permainan dan penerapan algoritma telah selesai
3. Iterasi kotak di arena lawan satu per satu
4. Apabila di kotak yang teriterasi tidak "hit", langsung menuju langkah 6. Apabila di kotak yang iterasi ternyata "hit", maka periksa kotak di sekeliling kotak

"hit" tersebut dimulai dari sebelah kanan, bawah, kiri, dan atas. Apabila terdapat "hit", maka hentikan iterasi di sekeliling kotak dan periksa kotak berikutnya dengan arah yang sama dengan sebelumnya sampai menemukan kotak yang tidak "hit".

5. Hitung berapa banyak kotak yang "hit" dan sesuaikan dengan ukuran dari kapal lawan.
6. Ulangi pendekatan dari langkah kedua

Untuk mempermudah penerapan algoritma *brute force* dalam simulasi, maka setiap jenis kapal akan disesuaikan dengan nomor yang diberikan kepadanya pada tabel 1 dan arena penempatan formasi kapal lawan juga akan disederhanakan menjadi array 1 dimensi berisikan class/type cell dengan 1 atribut boolean yang menyatakan bahwa dia ditempati sebuah kapal atau tidak, dan 1 atribut string berisikan "-" yang menandakan bahwa ia belum diiterasi, "O" yang menandakan bahwa ia telah diiterasi dan mendapat "hit", dan "X" yang menandakan bahwa ia telah diiterasi dan tidak mendapat "hit".

Dari langkah-langkah tersebut, maka dapat dihasilkan sebuah pseudocode sebagai berikut.

```

Type cell : record
    <isShip: boolean,
        IsIterated: boolean
    >

Function checkSurroundingCell(input i: int
, input/output A: array of cell) -> int
{Memeriksa sekeliling dari indeks i pada
array yang merupakan representasi arena
battleship dari horizontal maupun
vertikal. Fungsi mengembalikan ukuran
kapal yang ditenggelamkan.}

Function delFromArray(input i: int,
input/output A:array of int)
{Menghapus elemen array yang bernilai sama
dengan i}
    
```

Kamus Data:

```

eShip: array [0..5] of int
eArena: array [0..100] of cell
i, countHit: int
    
```

Algoritma:

{Diasumsikan array eArena telah tersedia dari studi kasus}

```

For i := 0 to 5 do
  {Mengisi daftar ship yang dimiliki lawan}
  eShip[i] <- i
i <- 0
while eShip.size > 0 and i < 100 do:
  if eArena[i].isShip and not eArena.isIterared then
    {Jika kotak yang diiterasi "hit"}
    countHit <- checkSurroundingCell(i, eArena)
    delFromArray(eShip, countHit)
  i++
  
```

Dengan pseudocode di atas, program akan mengiterasi seluruh kotak pada arena formasi kapal lawan. Hal ini memang merupakan penerapan dari algoritma *brute force* dengan pendekatan yang cukup sederhana, jelas, dan naif. Namun, kita dapat menerapkan metode heuristik untuk mengurangi iterasi yang dilakukan. Daripada mengiterasi seluruh kotak, kita dapat mengiterasi cukup hanya pada kotak yang diarsir kuning pada ilustrasi berikut.

	0	1	2	3	4	5	6	7	8	9
0	Yellow	White								
1	White	Yellow								
2	Yellow	White								
3	White	Yellow								
4	Yellow	White								
5	White	Yellow								
6	Yellow	White								
7	White	Yellow								
8	Yellow	White								
9	White	Yellow								

Tabel 4. Penerapan fungsi heuristik

Pada ilustrasi tabel di atas, kita menerapkan fungsi heuristik untuk mengiterasi kotak pada formasi kapal lawan. Fungsi heuristik yang digunakan adalah karena kapal dengan ukuran terkecil adalah Destroyer yang berukuran 2, maka jika satu kotak telah kita iterasi, maka kotak-kotak yang berada di sekelilingnya (kanan, bawah, kiri, atas) tidak perlu kita iterasi karena walaupun kotak-kotak tersebut merupakan sebuah kotak yang menaungi sebuah kapal, mereka pasti akan terdeteksi saat kita melakukan "hit" pada kotak bersirkon kuning lainnya. Misalkan pada contoh berikut.

	0	1	2	3
0	Yellow	White	Yellow	White

Tabel 5. Contoh Penerapan Metode Heuristik

Misalkan Destroyer merentang pada kotak [0,1] dan [0,2]. Saat kita mengiterasi kotak [0,0], iterasi berikutnya adalah kotak [0,2]. Dengan kata lain, kita melewati kotak [0,1] yang dinaungi kapal destroyer. Walaupun dilewati, saat kita berada pada kotak [0,2] dan mendapat "hit", kita akan memeriksa sekeliling dari kotak [0, 2] secara horizontal dan akan mendapatkan "hit" juga pada kotak [0, 1]. Oleh karena itu, fungsi heuristik ini dapat menghemat waktu berjalannya program. Dalam Pseudocode, dapat kita berikan modifikasi pada iterasi loop sebagai berikut.

```

If (i mod 10 = 8) then
  {Iterasi dari baris genap menuju baris ganjil}
  i = i + 3
else if (i mod 10 = 9) then
  {Iterasi dari baris ganjil menuju baris genap}
  i = i + 1
else
  {Iterasi tidak di ujung baris}
  i = i + 2
  
```

B. Data Pengujian Program

Dalam simulasi ini, penulis menggunakan 2 *test case* formasi kapal lawan sebagai berikut untuk membandingkan waktu pada penggunaan algoritma *brute force* pada permainan battleship.

	0	1	2	3	4	5	6	7	8	9
0										
1			Red	Red	Red	Red	Red			
2										
3	Red	Red	Red							
4				Red						
5										
6					Red	Red	Red	Red	Red	
7		Red								
8		Red								
9		Red								

Tabel 6. Test Case 1 formasi kapal lawan

	0	1	2	3	4	5	6	7	8	9
0	█		█					█	█	█
1	█		█							
2	█		█							
3	█		█							
4			█							
5										
6										
7										
8	█	█	█							
9									█	█

Tabel 7. Test Case 2 formasi kapal lawan

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5						█				
6						█		█		
7						█		█		
8	█	█	█	█	█					█
9				█	█	█	█			█

Tabel 7. Test Case 3 formasi kapal lawan

Program ini dijalankan dengan menggunakan bahasa pemrograman Python versi 3.7.4 pada laptop penulis yang memiliki spesifikasi sebagai berikut.

OS : Windows 10 dual boot Ubuntu 18.04 LTS
 Processor : AMD A9-9420 RADEON R5
 RAM : 4 GB

Dalam menjalankan program, penulis mendapatkan data dari masing-masing percobaan. Setiap *test case* diuji sebanyak lima kali untuk masing-masing tanpa dan dengan heuristik.

Test Case 1			
Tanpa Heuristik		Dengan Heuristik	
No.	Waktu (ms)	No.	Waktu (ms)
1	0.1461	1	0.1068
2	0.1532	2	0.1254
3	0.1354	3	0.1237
4	0.1419	4	0.1184
5	0.1424	5	0.1371
Rata-rata	0.1438	Rata-rata	0.1212

Tabel 8. Data pengujian *test case 1*

Test Case 2			
Tanpa Heuristik		Dengan Heuristik	
No.	Waktu (ms)	No.	Waktu (ms)
1	0.1479	1	0.1357
2	0.1852	2	0.1334
3	0.1773	3	0.1394
4	0.1232	4	0.1579
5	0.1546	5	0.1102
Rata-rata	0.1576	Rata-rata	0.1355

Tabel 9. Data pengujian *test case 2*

Test Case 3			
Tanpa Heuristik		Dengan Heuristik	
No.	Waktu (ms)	No.	Waktu (ms)
1	0.1567	1	0.1210
2	0.1467	2	0.1129
3	0.1449	3	0.1025
4	0.1410	4	0.1116
5	0.1421	5	0.1166
Rata-rata	0.1463	Rata-rata	0.1129

Tabel 10. Data pengujian *test case 3*

Dari data yang didapat, dapat dilihat bahwa penggunaan algoritma *brute force* pada permainan battleship dengan menerapkan fungsi heuristik membutuhkan waktu yang lebih

sedikit dibandingkan dengan algoritma *brute force* yang tidak menggunakan fungsi heuristik. Walaupun perbedaannya tidak terlihat signifikan, hal tersebut dikarenakan jumlah iterasi yang belum terlalu banyak sehingga perbedaannya tidak terlalu terlihat. Apabila jumlah iterasi yang dilakukan sudah sangat banyak, fungsi heuristik akan menghasilkan perbedaan yang signifikan pada penerapan algoritma *brute force* tersebut, tidak hanya pada permainan battleship.

IV. KESIMPULAN

Berdasarkan pengujian yang telah dilaksanakan, dapat dilihat bahwa penggunaan fungsi heuristik pada algoritma *brute force* pada permainan battleship dapat meningkatkan kecepatan waktu eksekusi dari program. Fungsi heuristik ini akan sangat berpengaruh pada permasalahan dengan jumlah iterasi yang banyak. Hal ini dapat menekankan bahwa sepintar apapun teknologi yang diciptakan, manusia akan tetap lebih pintar dari teknologi tersebut karena pikiran yang tertanam pada teknologi berasal dari pikiran manusia. Perlu diperhatikan bahwa algoritma yang digunakan penulis, baik yang menggunakan heuristik maupun tidak menggunakan heuristik, tidak menjamin kemenangan dalam permainan battleship.

VIDEO LINK AT YOUTUBE

Untuk memperjelas pembahasan dari isi makalah ini, penulis telah membuat video mengenai makalah ini yang dapat diakses pada link https://youtu.be/Bvfw_y9huPk

ACKNOWLEDGMENT

Penulis memanjatkan puji dan syukur kepada Tuhan Yang Maha Esa karena atas Rahmat dan karunia-Nya penulis dapat menyelesaikan masalah yang berjudul "Perbandingan Algoritma Brute Force dengan dan tanpa Fungsi Heuristik pada Permainan Battleship". Penulis ingin berterima kasih kepada ibu Dr. Masayu Leylia Khodra, S.T., M.T. selaku dosen mata kuliah IF2211 Strategi Algoritma yang telah membimbing saya selama keberjalanan kuliah ini. Tidak lupa penulis juga mengucapkan terima kasih kepada bapak Dr. Ir. Rinaldi Munir, M.T. dan ibu Dr. Nur Ulfa Maulidevi S.T., M.Sc yang juga merupakan dosen mata kuliah IF2211 Strategi Algoritma jurusan teknik informatika yang telah menyampaikan materi dengan baik walaupun di masa pandemik COVID-19 pada saat penulis membuat makalah ini. Penulis juga mengucapkan terimakasih kepada teman-teman teknik informatika Institut Teknologi Bandung angkatan 2018, khususnya teman-teman K-01. Penulis juga ingin berterima kasih kepada seluruh sumber referensi yang dicantumkan pada makalah ini sehingga makalah ini dapat mencantumkan informasi yang sebenar-benarnya.

REFERENCES

- [1] Munir, Rinaldi. Algoritma Brute Force. [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-\(2016\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-(2016).pdf). Diakses pada tanggal 2 Mei 2020, pukul 15.30
- [2] Dinand. Pengertian Algoritma Brute Force dan Greedy. <https://dosenit.com/ilmu-komputer/pengertian-algoritma-brute-force-dan-greedy>. Diakses pada tanggal 2 Mei pukul 15.35
- [3] Tim Penulis Fandom Board Games Galore. Battleship. <https://board-games-galore.fandom.com/wiki/Battleship>. Diakses pada tanggal 2 Mei 2020, pukul 16.30

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.

Jakarta, 3 Mei 2020



Ananda Yulizar Muhammad - 13518088