Penerapan Algoritma Brute Force pada Permainan Kotak Ajaib

Benardi Atmadja / 13510078
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13510078@std.stei.itb.ac.id

***Abstract*—Kotak ajaib merupakan suatu permainan matematika yang di mana sebuah matriks n x n diisi dengan bilangan dari 1 hingga n*n. Permainan ini dapat diselesaikan dengan berbagai cara penyelesaian algoritma. Pada makalah ini dijelaskan bagaimana algoritma BruteForce digunakan untuk menyelesaikan permasalahan kotak ajaib.**

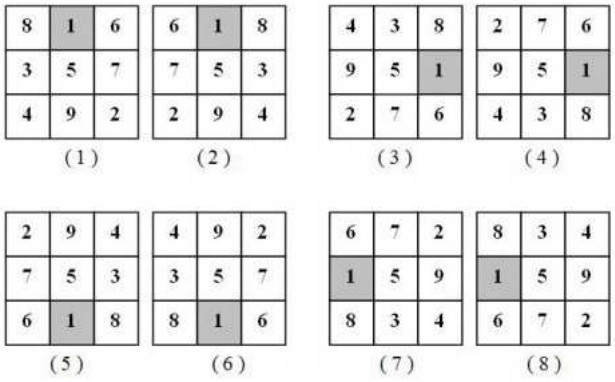
***Kata kunci*—Kotak ajaib, brute force.**

I. PENDAHULUAN

1. Kotak Ajaib

Permainan Kotak ajaib merupakan permainan matematika yang memerlukan logika. Pemain harus mengisi kotak-kotak yang kosong berdasarkan dengan menggunakan angka dari 1 hingga n x n. Misalnya pada kotak 3x3 maka angka yang tersedia adalah 1,2,3,4,5,6,7,8,9. Tujuan dari permainan ini adalah mengisi kotak dengan angka-angka yang tersedia dan tidak boleh dipakai dua kali, agar jumlah angka-angka pada kolom, baris, dan diagonal yang panjang sama.

Contoh penyelesaian pada kotak ajaib 3x3 ini adalah:



Gambar 1. Penyelesaian kotak ajaib 3x3

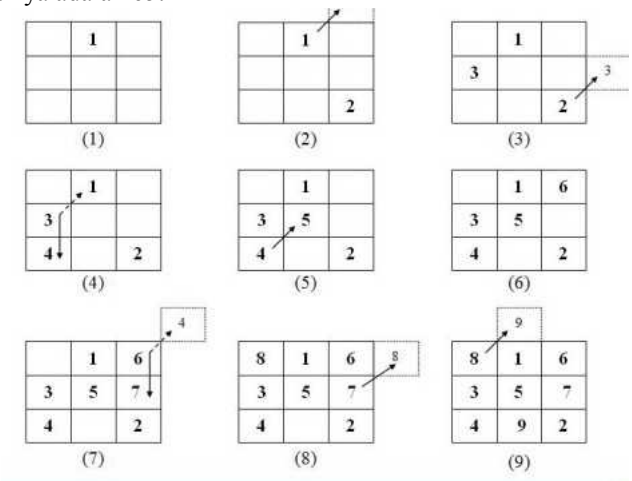
Solusi dari permainan ini ada 8 buah. Dapat dilihat bahwa bentuk yang ada sebenarnya berpola sama hanya saja posisi angka 1 dimulai pada sisi yang berbeda. Angka 1 pasti memiliki angka disampingnya yaitu 6 dan 8.

Pada matriks tersebut bisa dilihat bahwa jumlah kolom dan baris dan diagonal adalah 15.

Dengan menggunakan pendekatan *heuristic*, nilai jumlah 15 bisa didapatkan melalui penjumlahan deret angka yang mungkin dan membaginya dengan nilai n matriks. Contoh 1,2,3,4,5,6,7,8,9. Total penjumlahan dari deret tersebut adalah 1+2+3+4+5+6+7+8+9 = 45. Kemudian kita membaginya dengan n ukuran matriks yaitu 3 dan didapatkan nilai 15.

Cara lebih mudahnya adalah memanfaatkan nilai tengah dari kemungkinan angka tersebut. Nilai tengah adalah 5. Kemudian angka tersebut dikalikan dengan 3 sehingga hasilnya adalah 15.

Maka untuk matriks 5x5, nilai tengahnya adalah (1+25)/2 = 13. Kemudian dikalikan dengan n maka nilainya adalah 65.



Gambar 2. Penyelesaian kotak ajaib dengan menggunakan metode *heuristic*

(1) (15) (14) (4)
(10) (8) (5) (11)
(7) (9) (12) (6)
(16) (2) (3) (13)

Merupakan salah satu contoh penyelesaian dari kotak ajaib berukuran 4x4. Akan tetapi pada matriks 4x4 metode nilai tengah tidak bisa digunakan karena angka yang didapat adalah 34.

II. METODE

2. Algoritma Brute Force

Algoritma Brute Force merupakan algoritma yang paling tidak cerdas dan paling tidak mangkus dari algoritma lain yang ada. Hal ini dikarenakan algoritma brute force mencoba semua kemungkinan yang bisa dihasilkan dari suatu permasalahan. Dengan demikian semakin besar ukuran permasalahan, maka kemungkinan yang dihasilkan oleh algoritma ini akan menjadi semakin besar.

Akan tetapi algoritma Brute Force pada ukuran permasalahan yang sederhana dapat menyelesaikan permasalahan dengan cepat.

Algoritma Brute Force bekerja dengan mencocokkan setiap data L dengan DX yang didapatkan dari set X, apabila satu saja data yang dicocokkan tidak sesuai maka data L yang dimasukkan bukan merupakan hasil dari DX. Pada proses pengerjaan, algoritma ini terdiri dari empat tahap. Tahap pertama yaitu mengurutkan jsj dari array L (karena masukan nilai bisa acak), lalu algoritma ini mengkalkulasikan nilai set DX dari array X dengan cara mencari selisih dari isi nilai yang bernilai selain nol. Setelah itu algoritma ini akan menyusun nilai dari set DX yang terbentuk dari proses perhitungan sebelumnya, dan yang terakhir dari kedua array yang dibentuk di atas (DX dan L) akan dicocokkan nilainya satu dengan yang lain dan apabila ada satu saja nilai yang tidak sama maka akan keluar komentar "tidak ketemu".

Algoritma ini memiliki kompleksitas waktu dan ruang $O(M^{n2-n})$, ini disebabkan karena pada proses pencarian semua kemungkinan set (DX), algoritma ini mencoba semua selisih dari nilai yang ada di set X dan memasukkannya ke dalam array.

2.1 Pseudo – code Algoritma Brute Force

```
Procedure SortArray (L : array of integer)
Deklarasi
X : integer
Y : integer
Algoritma
X <- 0
While (L[X] /= nil) do
Y <- X + 1
Temp < - L[X]
While (L[Y] /= nil) do
If (L[Y] <= L[X]) then
Temp <- L[X]
L[X] <- L[Y]
L[Y] <- temp
Y++
X++

Function CreateDX (X: array of integer) -> L
Deklarasi
L : array of integer
K : integer
V : integer
J : integer
Algoritma
K <- 0
J <- 0
While (X[K] /= nil) do
V <- K + 1
While (X[V] /= nil) do
If (X[V] - X[K] /= 0) then
L[J] <- X[V] - X[K]
J++
V++
K++
  ------> L

Function VarianMatch (L : array of integer, X :
array of integer) -> X
{Prekondisi : jika array tidak cocok maka X akan
di- nil kan}
Deklarasi
DX : array of integer
J : integer
Algoritma
  SortArray (L)
  DX <- CreateDX(X)
  SortArray (DX)
  If (DX = L) then
  -----> X
  Else
  Write ("hasil tidak sama")
  X <- nil
  -----> X
```

2.3 Penerapan Algoritma Brute Force pada Kotak Ajaib

Buat suatu matriks n x n. Dari kotak awal (1,1) dimasukkan angka 1. Kemudian kotak selanjutnya 1,2 dicobakan angka 1. Karena peraturan pada permainan ini tidak boleh ada angka yang sama atau angka adalah unik, maka angka 1 tersebut akan diganti dengan angka berikutnya yaitu 2. Kemudian pada kotak 1,3 dicobakan angka dimulai lagi dengan angka 1. Karena angka 1 tidak bisa maka dicobakan angka berikutnya yaitu 2. Angka 2 sudah terpakai dan angka selanjutnya adalah 3. Maka kotak 1,3 diisi dengan nilai 3.

1 x x
x x x
x x x

Berdasarkan pendekatan heuristic yang sudah dijelaskan sebelumnya, nilai dari kotak tersebut harusnya adalah 15. Maka diperiksa apakah bilangan tersebut sudah berjumlah 15. Karena jumlah dari kotak tersebut adalah 6, maka angka tersebut pasti salah, jadi kita ganti bilangan 3 tadi dengan angka berikutnya yaitu 4.

1 2 3 1 2 4
x x x x x x
x x x x x x

Begitu seterusnya sampai angka 9 dan jumlah yang ada masih dibawah 15.

1 2 9
x x x
x x x

Sehingga perlu diganti angka sebelumnya yang tadinya 2 menjadi 3.

1 3 x
x x x
x x x

Perhitungan akan terus dilanjutkan ketika nilai baris adalah 15.

1 5 9
x x x
x x x

Dan bilangan ke 2,1 mulai diperiksa lagi dengan menggunakan angka yang bisa diletakkan.

1 5 9
2 x x
x x x

perhitungan terus dilakukan hingga didapat bilangan menjadi

1 5 9
2 6 7
X x x

Sekarang X besar salah karena bilangan masih dibawah 15 saat dicoba bilangan dari 1 – 9.

Maka dilakukan backtracking ke angka sebelumnya yaitu 7.

Angka 7 tidak bisa diubah karena jumlah baris adalah 15. Maka bilangan 6 yang coba diubah dengan 8. Namun kemungkinan bilangan pada 2,3 tidak boleh 5.

1 5 9
2 8 x
x x x

Perhitungan dilakukan mundur kembali hingga bilangan awal didapatkan 2.

2 x x
x x x
x x x

Penelusuran pada brute force dilakukan lagi hingga didapatkan solusi yang terdekat yaitu

2 7 6
9 5 1
4 3 8

Struktur data matriks dalam pembuatannya dapat dibuat dalam 2 cara.

Cara pertama menggunakan matriks 2 dimensi yaitu x dan y.

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

Gambar 3. Representasi matriks x dan y.

Setiap dilakukan pengecekan pada bilangan, maka perlu ditelusuri kotak-kotak sebelumnya apakah sudah diisi angka tersebut.

Apabila angka sudah mencapai ujung dari kolom atau baris. Maka dilakukan perhitungan penjumlahan menurut kolom dan baris.

Misalnya pada contoh di atas, Perhitungan kolom ke 1dilihat dari matriks [1][1 sampai n], yaitu [1][1], [1][2], dan [1],[3].

Dan penjumlahan pada kolom ke 1dilakukan dengan [1 sampai n][1], yaitu [1][1],[2][1],[3][1].

Pada penjumlahan diagonal, dilakukan dengan cara iterasi [n][n], yaitu [1][1],[2][2],[3][3]. Dan pada diagonal yang lain, dilakukan dengan cara iterasi naik dari [1 sampai n] dan iterasi turun [n sampai 1], yaitu [1][3],[2][2],[3][1].

Dengan demikian program Brute Force pada permainan kotak ajaib dapat menemukan jawaban yang benar.

III. KESIMPULAN

Algoritma Brute Force sangat tidak mangkus karena memerlukan penyelesaian yang sangat banyak dan tidak efisien namun dapat menyelesaikan permasalahan yang ada P

IV. REFERENSI

[1] Bujur Sangkar Ajaib (2009) http://h4mm4d.wordpress.com/2009/03/31/bujur-sangkar-ajaib/ Tanggal Pengaksesan (20 Desember 2012)

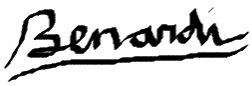
[2] Rinaldi. Munir, *Strategi Algoritma*, Teknik Informatika, Bandung, 2009.

V. 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 Desember 2012

ttd



Benardi Atmadja 13510078