

Penerapan Algoritma *Brute Force* di Permainan *Nonogram*

Aurelia 13512099
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13512099@itb.ac.id

Abstract—Algoritma *brute force* merupakan salah satu algoritma yang sering dipakai dan dapat diterapkan dalam banyak masalah. Hal ini dikarenakan *brute force* dapat menyelesaikan masalah dengan sangat sederhana, langsung, dan jelas. Salah satu penerapannya yang akan dibahas dalam makalah ini adalah dalam pencarian solusi permainan *Nonogram*. *Nonogram* adalah sejenis permainan puzzle logika dimana kita harus mengisi kotak menjadi warna hitam sesuai dengan petunjuk angka-angka yang tertulis di sisi-sisi kotak *Nonogram*, yang hasilnya akan membentuk sebuah gambar.

Index Terms—*brute force*, puzzle, kotak

I. PENDAHULUAN

Permainan sejenis puzzle yang bertujuan utama mengasah otak sudah sering sekali kita temui dalam kehidupan sehari-hari. Permainan seperti ini banyak ditemukan dalam bentuk yang berhubungan dengan kotak-kotak dan angka. Contoh permainan yang terkenal misalnya Sudoku.

Dalam makalah ini, akan dibahas sebuah permainan puzzle bernama *Nonogram*. Tujuan dari permainan ini adalah “mewarnai” kotak-kotak menjadi warna hitam sesuai dengan petunjuk yang diberikan melalui angka-angka yang terdapat disisi setiap baris dan kolom dari kotak *Nonogram*.

Salah satu cara penyelesaian masalah untuk pencarian solusi dari permainan ini adalah dengan algoritma *brute force*. Algoritma *brute force* memang algoritma yang mudah digunakan dan bisa dipastikan bahwa solusi didapatkan.

II. ALGORITMA BRUTE FORCE

Brute force adalah pendekatan yang lempang (*straightforward*) untuk memecahkan suatu masalah. Biasanya algoritma ini didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan. Algoritma *brute force* terkenal dengan pemecahan masalah yang sangat sederhana, langsung, dan jelas (*obvious way*). Oleh karena itu, algoritma *brute force* hampir selalu dapat memecahkan semua masalah yang dapat dipecahkan oleh algoritma lainnya.

Brute force dikenal sebagai algoritma yang paling

general dalam penyelesaian masalah. Algoritma ini berjalan dengan secara sistematis mencoba setiap cara yang memungkinkan untuk dijadikan solusi dan mengecek apakah setiap percobaan yang dijalankan sesuai dengan pernyataan masalah.

Contohnya, untuk mencari pembagi dari sebuah bilangan n , algoritma *brute force* akan mencoba semua kemungkinan bilangan bulat mulai dari 1 sampai ke nilai dari akar dari n . Selain contoh tersebut, ada berbagai contoh lain yang sering diselesaikan dengan algoritma *brute force*, yakni sebagai berikut.

- masalah pencarian elemen terbesar/terkecil
- pencarian beruntun
- menghitung nilai a^n
- menghitung nilai faktorial dari sebuah bilangan
- mengalikan dua buah matriks
- tes bilangan prima
- evaluasi polinom dan sebagainya.

Algoritma *brute force* sangat mudah diimplementasikan dan selalu menjamin ditemukannya solusi permasalahan.

Karakteristik Algoritma *Brute Force*

Seringkali algoritma *brute force* ini bersifat tidak “cerdas” dan tidak mangkus, dikarenakan cara kerjanya yang mencoba semua macam kemungkinan sehingga menghasilkan jumlah komputasi yang cukup besar untuk dapat menemukan solusi.

Algoritma *brute force* juga pada dasarnya lebih cocok untuk masalah yang berukuran kecil. Untuk masalah yang terlalu luas dan kompleks, *brute force* akan menjadi semakin tidak mangkus. *Brute force* sering digunakan sebagai basis pembanding dengan algoritma yang lebih mangkus.

Oleh karena itu, hampir semua permasalahan yang dapat diselesaikan dengan algoritma lain, hampir sudah pasti dapat pula diselesaikan dengan algoritma *brute force*.

Kelebihan *Brute Force*

Dari penjelasan karakteristik diatas, dapat dilihat bahwa salah satu kelebihan *brute force* yang paling terlihat adalah

bahwa algoritma ini dapat digunakan untuk memecahkan hampir sebagian masalah, dan penerapannya dapat menjadi sangat luas.

Selain itu, dibanding semua algoritma lainnya, algoritma brute force dapat dikatakan yang paling sederhana dan paling mudah dimengerti. Algoritma brute force menghasilkan algoritma yang layak untuk beberapa masalah penting seperti pencarian, pengurutan, pencocokan, perkalian matriks, dan sebagainya.

Algoritma yang dihasilkan brute force baku (standard) untuk beberapa pekerjaan penghitungan semisal penjumlahan/perkalian n buah bilangan, menentukan elemen minimum atau maksimum dalam sebuah tabel.

Kelemahan Brute Force

Untuk beberapa masalah yang kompleks, algoritma brute force sulit untuk menghasilkan solusi yang mangkus. Beberapa algoritma brute force juga sangat lambat sehingga tidak dapat diterima. Selain itu, brute force tidak sekonstruktif dan sekreatif algoritma-algoritma penyelesaian masalah lainnya.

Exhaustive Search

Exhaustive search adalah teknik pencarian solusi secara brute force untuk masalah-masalah kombinatorik. Biasanya di antara objek-objek kombinatorik seperti permutasi, kombinasi, atau himpunan bagian dari sebuah himpunan.

Langkah-langkah dari exhaustive search adalah sebagai berikut.

1. Enumerasi setiap solusi yang mungkin dengan cara yang sistematis.
2. Evaluasi setiap kemungkinan solusi satu per satu, simpan solusi terbaik yang ditemukan sampai sejauh ini.
3. Bila pencarian berakhir, umumkan solusi terbaik.

Contoh-contoh exhaustive search adalah sebagai berikut.

- Travelling Salesperson Problem
- 1/0 Knapsack
- Kriptografi

Contoh Penerapan Brute Force

Algoritma brute force dapat diterapkan dalam masalah pencocokan string (*string matching*).

Persoalan yang perlu diselesaikan adalah sebagai berikut.

Diberikan sebuah teks, yaitu *string* dengan panjang n karakter, dan sebuah *pattern*, yaitu *string* dengan panjang m karakter dengan asumsi $m < n$. Tujuannya adalah mencari lokasi pertama di dalam teks yang bersesuaian dengan

pattern.

Algoritma brute force yang dipakai :

1. Mula-mula *pattern* dicocokkan pada awal teks.
2. Dengan bergerak dari kiri ke kanan, bandingkan setiap karakter di dalam *pattern* dengan karakter yang bersesuaian di dalam teks sampai :
 - semua karakter yang dibandingkan cocok atau sama (pencarian berhasil), atau
 - dijumpai sebuah ketidakcocokan karakter (pencarian belum berhasil)
3. Bila *pattern* belum ditemukan kecocokannya dan teks belum habis, geser *pattern* satu karakter ke kanan dan ulangi langkah 2.

Contoh :

Pattern : RITMA

Teks : ALGORITMA BRUTE FORCE

ALGORITMA BRUTE FORCE

1. RITMA
2. RITMA
3. RITMA
4. RITMA
5. **RITMA**

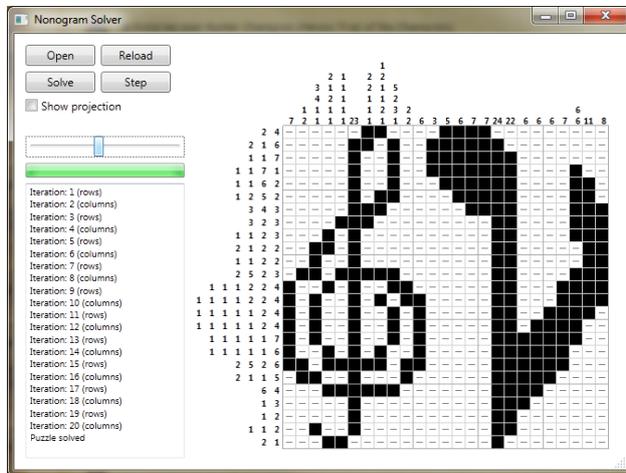
Algoritmanya adalah sebagai berikut.

```
procedure PencocokanString(input P : string, T : string,
                           n, m : integer, output idx : integer)
{ Masukan: pattern P yang panjangnya m dan teks T yang
  panjangnya n. Teks T direpresentasikan sebagai string
  (array of character)
  Keluaran: lokasi awal kecocokan (idx)
}
Deklarasi
  i : integer
  ketemu : boolean
Algoritma:
  i ← 0
  ketemu ← false
  while (i ≤ n-m) and (not ketemu) do
    j ← 1
    while (j ≤ m) and (Pj = Ti+j) do
      j ← j+1
    endwhile
    { j > m or Pj ≠ Ti+j }
    if j = m then { kecocokan string ditemukan }
      ketemu ← true
    else
      i ← i+1 {geser pattern satu karakter ke kanan teks }
    endif
  endfor
  { i > n - m or ketemu }
  if ketemu then
    idx ← i+1
  else
    idx ← -1
  endif
```

Kompleksitas dari algoritma tersebut adalah $O(nm)$ untuk kasus terburuk dan $O(n)$ untuk kasus rata-rata.

III. PERMAINAN NONOGRAM

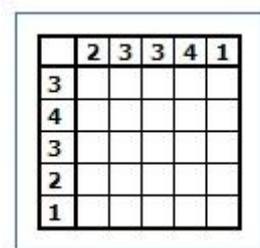
Nonogram adalah permainan teka-teki sederhana yang ditemukan oleh seseorang berkewarganegaraan Jepang pada tahun 1987. Saat ini, sudah banyak nama yang dipakai untuk menggantikan nama Nonogram, misalnya Paint by Numbers, Griddlers, Picross, Hanjie, dan masih banyak lagi. Sampai sekarang, Nonogram sudah dimainkan dalam berbagai bentuk, mulai dari media cetak maupun elektronik.



Gambar 1. Contoh Nonogram Solver

Nonogram ini terdiri dari kotak-kotak yang membentuk tabel yang ukurannya bervariasi, mulai dari 5x5, 10x10, dan sebagainya.

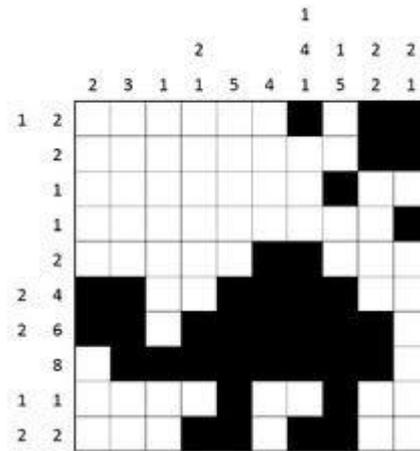
Di awal permainan, papan permainan akan terlihat sebagai berikut.



Gambar 2. Tampilan awal papan permainan

Gambar diatas merupakan papan permainan Nonogram untuk ukuran 5x5. Terdapat 25 kotak putih yang nantinya akan kita isi dengan warna hitam sesuai dengan petunjuk angka yang diberikan. Petunjuk angka yang dimaksud adalah angka-angka yang berada di sisi kiri setiap baris dan sisi atas setiap kolom dari kotak-kotak Nonogram. Misalnya, untuk kolom pertama di gambar diatas, ada angka 2 yang menunjukkan bahwa kotak di kolom tersebut memiliki jumlah 2 kotak untuk diwarnai menjadi hitam. Begitu juga dengan kolom lain, dan hal yang sama berlaku untuk setiap baris.

Ada juga petunjuk angka yang berisi lebih dari satu angka, misalnya "4 8 2". Itu berarti di kolom/baris tersebut harus ada kelompok kotak yang berwarna hitam sejumlah 4, 8, dan 2, dan masing-masing kelompok harus terpisah oleh minimal 1 kotak putih (yang tidak diwarnai).



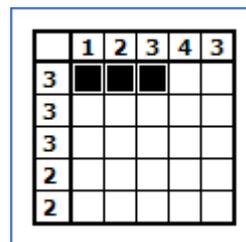
Gambar 3. Contoh lain Nonogram yang sudah selesai

Untuk menyelesaikan permainan ini, kita harus menentukan kotak mana yang akan diwarnai menjadi hitam dan mana yang tidak. Biasanya digunakan tanda silang untuk menandai kotak yang akan dibiarkan kosong. Dalam menentukannya, tidak dapat hanya menggunakan metode menebak, karena harus diperhitungkan angka-angka yang menjadi petunjuk.

IV. PENYELESAIAN NONOGRAM DENGAN BRUTE FORCE

Penyelesaian masalah permainan Nonogram dengan brute force ini adalah dengan berdasarkan konsep utama brute force, yakni mencoba semua kemungkinan yang ada. Penjelasan algoritmanya adalah sebagai berikut.

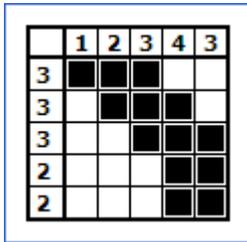
1. Pertama-tama, pilih bagian mana yang akan diselesaikan terlebih dahulu, bagian baris atau kolom.
2. Lakukan pewarnaan sesuai dengan petunjuk angka yang ada. Pewarnaan dapat dimulai dari kolom paling kiri dan baris paling atas.



Gambar 4. Contoh pewarnaan awal.

3. Lakukan pengecekan terhadap pewarnaan yang sudah dilakukan. Jika terdapat kesalahan, misalnya tidak sesuai dengan petunjuk angka yang ada, maka lakukan penggeseran pewarnaan sebanyak 1 kotak. Jika sudah mencapai ujung baris/kolom, maka ulangi pengaturan pewarnaan kotak-kotak yang sudah diwarnai sebelumnya.

4. Ulangi langkah pewarnaan dan pengecekan sampai tidak ditemukan kesalahan ketika pengecekan.



Gambar 5. Solusi permainan ditemukan.

5. Solusi untuk permainan ini sudah dapat ditemukan.

Program penyelesaian Nonogram dengan brute force dijelaskan sebagai berikut.

Pertama-tama, lakukan inialisasi dari papan permainan awal Nonogram, misalnya sebuah matriks berukuran 5x5 yang awalnya kosong. Algoritma brute force akan melakukan pengecekan dengan mengganti dan menggeser kotak satu per satu, sampai ditentukan bahwa solusi benar dan sesuai dengan pernyataan masalah.

Fungsi yang digunakan untuk mengubah kotak yakni mengecek apakah pewarnaan dapat digeser. Jika ya, maka akan digeser sebanyak satu kotak. Sebaliknya, lakukan pemindahan warna ke ujung paling kiri yang mungkin. Lalu ulangi fungsi untuk pewarnaan selanjutnya.

Contoh kode penyelesaian Nonogram dengan algoritma brute force.

```
public static void main (String [] args)
{
    int [] [] grid = {
        {1, 1, 5, 1},
        {1, 5, 3, 1},
        {1, 4, 1, 1},
        {1, 1, 1, 1},
    };

    System.out.println ("The original array");

    for (int row = 0 ; row < grid.length ; row++)
    {
        for (int col = 0 ; col < grid [0].length ; col++)
        {
            System.out.print (grid [row] [col] + " ");
        }
        System.out.println ();
    }
}
```

```

}

System.out.println ("The new array");
int currentrow = 2;
int currentcol = 2;

boolean forward = true;
boolean carryOn = true;
boolean solution = true;
while (carryOn)
{
    try
    {
        System.out.println(grid[2][2]);
        //move forward to end of array
        if (forward)
        {
            if (currentcol == grid.length)
            {
                currentcol = 0;
                currentrow++;
            }
        }

        //move backward to front of array
        if (!forward)
        {
            if ((currentcol < 0) && (currentrow
!=0))
            {
                currentcol = grid [0].length - 1; //not
for first row
                currentrow--;
            }
        }

        //if gridvalue is not 1
        if (grid [currentrow] [currentcol] != 1)
        {
            //add 1 to value
            grid [currentrow] [currentcol] += 1;
            forward = true;
        }
        //if gridvalue = 1
        else
        {
            //set gridvalue to -1
            grid [currentrow] [currentcol] = -1;
            forward = false;
            //Now to move one cell back
            currentcol -= 1;
            //System.out.println (currentcol);
        }
        if (forward)
        {
            currentcol++;
        }
        //when end of array is reached
        if ((currentrow == grid.length - 1) &&
((currentcol == grid.length)))
    }
}

```

```

        {
            carryOn = false;
        }
    }
    catch (Exception err)
    {
        System.out.println ("No solution exists!");
        carryOn = false;
        solution = false;
    }
}

if (solution)
{
    for (int row = 0 ; row < grid.length ; row++)
    {
        for (int col = 0 ; col < grid [0].length ;
col++)
        {
            System.out.print (grid [row] [col] + " ");
        }
        System.out.println ();
    }
}

} // main method

```

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, 19 Mei 2014

ttd



Aurelia 13512099

V. KESIMPULAN

Nonogram merupakan permainan puzzle untuk mengasah otak, yang melibatkan kotak-kotak tabel dan angka-angka yang menjadi petunjuk. Permainan ini cukup menyenangkan untuk dimainkan. Salah satu cara penyelesaian permainan ini yang dapat diterapkan adalah dengan algoritma brute force. Algoritma brute force memang terkenal sebagai algoritma yang dapat memecahkan hampir berbagai macam masalah dikarenakan sifatnya yang sederhana dan mudah diimplementasikan serta terjamin didapatkannya solusi penyelesaian. Walaupun demikian, algoritma brute force, misalnya dalam penyelesaian masalah permainan Nonogram ini, bukanlah merupakan algoritma terbaik yang dapat dipakai, sebab algoritma ini memang tidak mangkus untuk digunakan dalam masalah yang lingkupnya cukup luas dan kompleks. Oleh karena itu, sebenarnya permainan Nonogram dapat diselesaikan dengan algoritma lain yang lebih mangkus daripada brute force.

REFERENCES

- [1] [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2013-2014-genap/Algoritma%20Brute%20Force%20\(2014\).ppt](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2013-2014-genap/Algoritma%20Brute%20Force%20(2014).ppt)
- [2] <http://en.wikipedia.org/wiki/Nonogram>