

# Analisis Beberapa Algoritma Untuk Menyelesaikan Puzzle Sudoku

Yudha Setia RGP<sup>1</sup>, Ditto Narapratama<sup>2</sup>, Amahl AbdalKarim Salim<sup>3</sup>

Laboratorium Ilmu dan Rekayasa Komputasi  
Departemen Teknik Informatika, Institut Teknologi Bandung  
Jl. Ganesha 10, Bandung

E-mail : [if14118@students.if.itb.ac.id](mailto:if14118@students.if.itb.ac.id)<sup>1</sup>, [if14132@students.if.itb.ac.id](mailto:if14132@students.if.itb.ac.id)<sup>2</sup>,  
[if14140@students.if.itb.ac.id](mailto:if14140@students.if.itb.ac.id)<sup>3</sup>

## Abstrak

Permainan Sudoku adalah salah satu puzzle yang paling banyak digemari saat ini, dan juga merupakan salah satu permasalahan paling sulit di bidang informatika. Permasalahan puzzle Sudoku sulit untuk dipecahkan karena masuk dalam permasalahan *NP-complete*, sehingga tidak bisa diselesaikan dalam waktu polinomial. Hingga saat ini banyak programmer yang mencari algoritma yang mangkus untuk menyelesaikan puzzle ini. Cara yang paling naif adalah *Brute Force* yaitu dengan cara mengenumerasikan semua kemungkinan isi sel dengan angka 1 sampai 9. Tetapi cara ini tentu saja tidak mangkus karena kemungkinannya akan sangat banyak sekali. Karena itu algoritma ini diperbaiki dengan menambahkan *constraints* atau batasan, yaitu tidak boleh ada angka yang sama dalam satu baris, kolom atau *subgrid*. Cara ini bisa mereduksi jumlah kemungkinan secara signifikan sehingga algoritma menjadi lebih mangkus.

**Kata kunci:** *Sudoku, NP-Complete, Brute Force, Constraint Programming*

## 1. Pendahuluan

Permainan Sudoku adalah permainan yang dapat melatih logika manusia dalam berpikir cepat dan teliti. Permainan ini tidak bisa sembarang dimainkan, karena bila kita bermain dengan sembarangan di awal permainan, kita tidak bisa menyelesaikan game ini.

Umumnya Sudoku diselesaikan dengan cara manual, yaitu dengan cara mengeliminasi kemungkinan pengisian sel. Akan tetapi, karena rasa ingin tahu para programmer-programmer ketika menemukan masalah yang baru, pemecahan game sudoku ini menjadi salah satu bahan kajian yang bisa dibilang cukup rumit. Karena alasan itulah kami mengambil bahan kajian game sudoku ini menjadi bahan makalah kami.

Tujuan penulisan makalah ini adalah :

1. Menyelesaikan tugas yang diberikan oleh Dosen Mata Kuliah IF2251
2. Mempopulerkan game sudoku di kalangan mahasiswa informatika ITB
3. Membandingkan algoritma *brute force* murni dengan *brute force* dengan *constraint*

## 2. Ruang lingkup

Ruang lingkup pembahasan makalah kali ini adalah algoritma brute force murni dan brute force dengan constrain dalam menyelesaikan game sudoku

## 3. Sekilas Tentang Sudoku

Permainan Sudoku pertama kali muncul pada tahun 1979 di majalah Dell Magazines dengan nama "Number Places", kemudian menjadi populer dengan nama Sudoku di jepang pada tahun 1984. Puzzle Sudoku terdiri dari matriks 9x9 yang memiliki 9 sub-matriks berukuran 3x3 yang disebut subgrid. Tujuan dari permainan ini adalah mengisi semua sel dengan angka 1 sampai 9 sedemikian hingga setiap kolom, baris, dan subgrid mengandung angka 1 sampai 9 tepat satu buah. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini :

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 6 |   |   |   | 5 | 8 | 4 | 9 |   |
|   | 2 | 4 |   |   |   |   |   |   |
|   |   |   |   | 2 |   |   | 3 | 6 |
|   |   |   |   |   |   | 9 |   | 7 |
| 7 |   |   | 3 |   |   | 8 |   |   |
| 1 | 5 |   |   | 8 | 9 |   |   |   |
|   | 3 | 1 |   |   |   |   |   |   |
|   |   |   |   | 3 |   | 5 |   |   |
|   |   | 8 |   | 9 | 5 |   |   |   |

Sudoku yang belum diselesaikan

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 6 | 7 | 3 | 1 | 5 | 8 | 4 | 9 | 2 |
| 9 | 2 | 4 | 6 | 7 | 3 | 1 | 5 | 8 |
| 8 | 1 | 5 | 9 | 2 | 4 | 7 | 3 | 6 |
| 3 | 8 | 6 | 5 | 4 | 2 | 9 | 1 | 7 |
| 7 | 4 | 9 | 3 | 1 | 6 | 8 | 2 | 5 |
| 1 | 5 | 2 | 7 | 8 | 9 | 6 | 4 | 3 |
| 5 | 3 | 1 | 4 | 6 | 7 | 2 | 8 | 9 |
| 2 | 9 | 7 | 8 | 3 | 1 | 5 | 6 | 4 |
| 4 | 6 | 8 | 2 | 9 | 5 | 3 | 7 | 1 |

Sudoku yang sudah diselesaikan

#### 4. Beberapa cara menyelesaikan Sudoku

Cara normal untuk menyelesaikan puzzle Sudoku tanpa bantuan komputer adalah dengan menulis semua angka yang mungkin di setiap sel. Lalu menyelesaikan Sudoku dengan beberapa teknik-teknik yang sudah dikenal. Berikut adalah salah satu contoh menyelesaikan puzzle Sudoku tanpa bantuan komputer :

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
|   | 5 |   | 6 |   |   |   |   |   |
| 7 | 8 |   | 7 | 8 | 9 |   | 7 | 8 |
| 4 |   | 2 |   | 6 |   |   | 3 |   |
| 1 | 2 |   | 2 |   |   | 1 | 2 |   |
| 5 |   |   |   |   |   |   |   |   |
| 8 |   |   | 8 | 9 |   |   | 8 |   |
| 2 |   | 1 |   |   |   | 2 |   |   |
| 7 |   |   |   |   |   | 7 | 9 |   |
| 6 |   | 2 | 3 |   |   |   | 4 |   |
|   |   | 7 | 8 |   |   |   |   |   |
|   |   | 5 |   |   |   |   |   |   |
| 7 | 8 |   |   |   |   | 7 | 8 | 9 |
| 3 | 4 |   |   |   |   |   | 6 |   |
| 9 |   |   |   |   |   |   | 5 |   |
| 1 | 2 |   | 2 |   |   | 1 | 2 |   |
| 7 | 8 |   | 7 | 8 |   |   | 7 | 8 |

Contoh di sebelah kiri adalah potongan puzzle Sudoku. Sel yang kosong sudah diisi semua angka yang mungkin diisi. Dapat dilihat bahwa pada solusi untuk kotak berwarna kuning adalah angka 3, karena setiap kolom harus mengandung angka 3.

Cara menyelesaikan Sudoku dengan bantuan komputer adalah dengan cara mengimplementasikan algoritma. Algoritma yang bisa dipakai untuk menyelesaikan Sudoku diantaranya adalah dengan cara Brute Force murni, Brute Force dengan *constraint*, dan algoritma *Dancing Links*. Algoritma *Dancing Links* tidak dibahas di makalah ini karena algoritma ini di luar materi kuliah kami.

#### 5. Penyelesaian dengan Brute Force Murni

Cara paling naif adalah dengan cara mencoba semua kemungkinan pengisian sel dengan angka 1 sampai 9 tanpa mematuhi aturan Sudoku. Jadi dengan algoritma ini semua kemungkinan dicoba termasuk kasus yang menyalahi aturan seperti terdapat angka yang sama dalam satu baris/kolom/*subgrid*. Dengan begitu jumlah kemungkinan pengisian sel sebanyak  $9^{81}$  buah, atau sekitar  $1,9 \times 10^{77}$  buah. Tentu saja cara ini sangat tidak mangkus, karena kasus terburuk (*worst case*) bila menggunakan cara ini adalah solusi ditemukan pada enumerasi ke  $9^{81}$ , dapat disimpulkan teknik ini sangat lambat jadi dibutuhkan algoritma yang lebih mangkus dari Brute Force murni.

#### 6. Penyelesaian dengan Brute Force dengan *constraint*

Bila menggunakan *brute force* murni, kemungkinan yang meyalahi aturan juga dicoba. Tentu saja ini perbuatan yang tidak perlu sehingga kemungkinan yang menyalahi aturan perlu dieliminasi. Algoritma *Brute Force* dengan *constraint* ini mengeliminasi semua kemungkinan yang menyalahi aturan yaitu :

- setiap angka di tiap baris harus unik
- setiap angka di tiap kolom harus unik
- setiap angka di tiap *subgrid* harus unik

Menurut paper yang diterbitkan oleh Felgenhauer dan Jarvis, terdapat 6.670.903.752.021.072.936.960 ( $6,67 \times 10^{21}$ ) kemungkinan cara pengisian Sudoku yang sesuai dengan aturan. Dapat dilihat bahwa jumlah kemungkinan solusi jauh berkurang dibandingkan dengan algoritma *brute force* murni. Oleh karena itu cara ini lebih mangkus dan sering dipakai untuk menyelesaikan Sudoku.

## 7. Kesimpulan

Dalam penyelesaian game sudoku ini, dapat disimpulkan bahwa perbedaan antara *brute force* murni dengan *brute force* dengan *constrain* sangat jauh. Dari hasil yang didapat, algoritma *brute force* dengan *constrain* jauh lebih mangkus daripada menggunakan algoritma *brute force* murni.

## Daftar Pustaka

[http://en.wikipedia.org/wiki/Mathematics\\_of\\_Sudoku](http://en.wikipedia.org/wiki/Mathematics_of_Sudoku)

[u](http://en.wikipedia.org/wiki/Sudoku)

<http://en.wikipedia.org/wiki/Sudoku>

Rinaldi Munir, Diktat Kuliah Matematika  
Diskrit. Departemen Teknik Informatika  
ITB : 2005.