

Program Game *Hyper-Sudoku* dengan Algoritma

Brute Force

Johan - 13514026

Hyper-sudoku adalah varian Sudoku yang paling terkenal. Ia dikenal juga dengan nama "*NRC Sudoku*", "*Windoku*", dan "*4 Square Sudoku*". *Layout* permainan ini mirip dengan sudoku, tetapi dengan penambahan persegi 3 x 3 di dalamnya (diberi arsiran berwarna ungu) dengan syarat semua angka 1 sampai 9 harus muncul di dalam kotak persegi itu.

Cara bermainnya sedikit berbeda dengan sudoku biasa, karena penekanannya pada kotak persegi di dalamnya. Kotak persegi memberikan banyak informasi kepada pemain untuk, secara logika, mengurangi kemungkinan pengisian nilai pada kotak-kotak lain yang tersisa. (Sumber: Wikipedia)

Kotak-kotak di dalam hyper-sudoku harus diisi dengan angka 1 sampai 9 sedemikian sehingga:

1. tidak ada angka yang sama (berulang) pada setiap baris;
2. tidak ada angka yang sama (berulang) pada setiap kolom;
3. tidak ada angka yang sama (berulang) pada setiap bujursangkar (persegi) yang lebih kecil.
4. kotak persegi berwarna ungu berisi angka 1 sampai 9

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

Kondisi awal *Hypersudoku*

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

Solusi *Hypersudoku*

Dalam program penyelesaian Sudoku ini, dipergunakan Algoritma *Brute Force*. Algoritma yang digunakan adalah sebagai berikut :

1. Menulis angka "1" pada cell pertama yang kosong. Kemudian periksa apakah penempatan angka tersebut memenuhi syarat hyper-sudoku
2. Jika diperbolehkan,, maka maju ke cell selanjutnya yang masih kosong.
3. Tempatkan kembali angka "1" pada cell tersebut dan kemudian, periksa kembali apakah memenuhi syarat.
4. Jika tidak memenuhi syarat, berarti angka "1" tidak diperbolehkan. Sehingga coba dengan angka lainnya yaitu "2" atau "3" atau "4" dan seterusnya sampai 9.
5. Jika penempatan angka tidak ada yang memenuhi satupun angka dari 1-9, maka, tinggalkan cell tersebut, dan mundur ke cell yang sebelumnya diisi. Nilai di cell tersebut diganti dengan nilai yang mungkin dan memenuhi syarat.

6. Ulangi langkahnya hingga 81 cell terisi semua dengan benar
7. Jika hingga terakhir masih terdapat cell kosong tetapi tidak bisa diisi dengan angka 1-9, berarti Sudoku tersebut tidak memiliki solusi

Berikut adalah *source code* dari program *sudoku solver* yang dibikin

Kelas memory

```
package hypersudoku;

public class memory {

    int brs;

    int kol;

    public memory (int brs, int kol){

        this.brs = brs;

        this.kol = kol;

    }

}
```

Kelas matriks

```
package hypersudoku;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Scanner;

public class matriks {

    //atribut
    int [ ] [ ] Matriks = new int [9][9];

    public int count =0;

    void isimatriks()
    {
        try{
            int i, j;

            Scanner in = new Scanner (new File ("sudoku.txt"));
            for (i=0; i<9; i++) {
                for (j=0; j<9; j++) {
                    if(in.hasNextInt()){
                        this.Matriks[i][j] = in.nextInt();
                    }
                }
            }
            in.close();
        }
        catch (FileNotFoundException ex) {
            System.err.println("Error FNF" + ex.getMessage());
        }
        catch (IOException ex) {
            System.err.println("Error IO" + ex.getMessage());
        }
    }

}
```

```

void printmatriks()
{
    int i, j;

    System.out.println("+-----+-----+-----+");
    for(i=0; i<9; i++)
    {
        for(j=0; j<9; j++)
        {
            System.out.print("|" + Matriks[i][j]);

        }
        System.out.println("|");
        if (i%3 == 2) System.out.println("+-----+-----+-----+");
    }
}

```

```

public boolean isfilled (memory m)
{
    for (int i=0; i<9; i++)
    {
        for (int j=0; j<9; j++)
        {
            if (Matriks[i][j] == 0)
            {
                m.brs = i;
                m.kol = j;
                return true;
            }
        }
    }
    return false;
}

```

```

public boolean cekbrs (int brs, int nilai)
//mengecek apakah dalam satu baris terdapat angka yang sama dengan nilai yang akan diassign
{
    int j; //kolom
    for (j=0; j<9; j++)
    {
        if (this.Matriks[brs][j] == nilai){
            return false;
        }
    }
    return true;
}

```

```

public boolean cekkol (int kol, int nilai)
//mengecek apakah dalam satu kolom terdapat angka yang sama dengan nilai yang akan diassign
{
    int i; //baris

    for(i=0; i<9; i++)
    {
        if (this.Matriks[i][kol] == nilai){
            return false;
        }
    }
    return true;
}

```

```
int nokotak(int brs, int kol)
```

```
{
    if (brs>=0 && brs<9 && kol>=0 && kol<9)
    {
        if (brs>=0 && brs <3 && kol >= 0 && kol <3)
            return 1;
        if (brs>=0 && brs <3 && kol >= 3 && kol <6)
            return 2;
        if (brs>=0 && brs <3 && kol >= 6 && kol <9)
            return 3;
        if (brs>=3 && brs <6 && kol >= 0 && kol <3)
            return 4;
        if (brs>=3 && brs <6 && kol >= 3 && kol <6)
            return 5;
        if (brs>=3 && brs <6 && kol >= 6 && kol <9)
            return 6;
        if (brs>=6 && brs <9 && kol >= 0 && kol <3)
            return 7;
        if (brs>=6 && brs <9 && kol >= 3 && kol <6)
            return 8;
        if (brs>=6 && brs <9 && kol >= 6 && kol <9)
            return 9;
    }
    return 0;
}
```

```
boolean cekbox (int nokotak, int nilai)
```

```
{
    switch (nokotak) {
        case 0 : System.out.println("Error kotak");
        case 1 : return this.Matriks[0][0] != nilai && this.Matriks[0][1] != nilai && this.Matriks[0][2] != nilai &&
            this.Matriks[1][0] != nilai && this.Matriks[1][1] != nilai && this.Matriks[1][2] != nilai &&
            this.Matriks[2][0] != nilai && this.Matriks[2][1] != nilai && this.Matriks[2][2] != nilai;
        case 2 : return this.Matriks[0][3] != nilai && this.Matriks[0][4] != nilai && this.Matriks[0][5] != nilai &&
            this.Matriks[1][3] != nilai && this.Matriks[1][4] != nilai && this.Matriks[1][5] != nilai &&
            this.Matriks[2][3] != nilai && this.Matriks[2][4] != nilai && this.Matriks[2][5] != nilai;
        case 3 : return this.Matriks[0][6] != nilai && this.Matriks[0][7] != nilai && this.Matriks[0][8] != nilai &&
            this.Matriks[1][6] != nilai && this.Matriks[1][7] != nilai && this.Matriks[1][8] != nilai &&
            this.Matriks[2][6] != nilai && this.Matriks[2][7] != nilai && this.Matriks[2][8] != nilai;
        case 4 : return this.Matriks[3][0] != nilai && this.Matriks[3][1] != nilai && this.Matriks[3][2] != nilai &&
            this.Matriks[4][0] != nilai && this.Matriks[4][1] != nilai && this.Matriks[4][2] != nilai &&
            this.Matriks[5][0] != nilai && this.Matriks[5][1] != nilai && this.Matriks[5][2] != nilai;
        case 5 : return this.Matriks[3][3] != nilai && this.Matriks[3][4] != nilai && this.Matriks[3][5] != nilai &&
            this.Matriks[4][3] != nilai && this.Matriks[4][4] != nilai && this.Matriks[4][5] != nilai &&
            this.Matriks[5][3] != nilai && this.Matriks[5][4] != nilai && this.Matriks[5][5] != nilai;
        case 6 : return this.Matriks[3][6] != nilai && this.Matriks[3][7] != nilai && this.Matriks[3][8] != nilai &&
            this.Matriks[4][6] != nilai && this.Matriks[4][7] != nilai && this.Matriks[4][8] != nilai &&
            this.Matriks[5][6] != nilai && this.Matriks[5][7] != nilai && this.Matriks[5][8] != nilai;
        case 7 : return this.Matriks[6][0] != nilai && this.Matriks[6][1] != nilai && this.Matriks[6][2] != nilai &&
            this.Matriks[7][0] != nilai && this.Matriks[7][1] != nilai && this.Matriks[7][2] != nilai &&
            this.Matriks[8][0] != nilai && this.Matriks[8][1] != nilai && this.Matriks[8][2] != nilai;
        case 8 : return this.Matriks[6][3] != nilai && this.Matriks[6][4] != nilai && this.Matriks[6][5] != nilai &&
            this.Matriks[7][3] != nilai && this.Matriks[7][4] != nilai && this.Matriks[7][5] != nilai &&
            this.Matriks[8][3] != nilai && this.Matriks[8][4] != nilai && this.Matriks[8][5] != nilai;
        case 9 : return this.Matriks[6][6] != nilai && this.Matriks[6][7] != nilai && this.Matriks[6][8] != nilai &&
            this.Matriks[7][6] != nilai && this.Matriks[7][7] != nilai && this.Matriks[7][8] != nilai &&
            this.Matriks[8][6] != nilai && this.Matriks[8][7] != nilai && this.Matriks[8][8] != nilai;
        default: return false;
    }
}
```

```
int nokotakhyper(int brs, int kol)
```

```
{
    if (brs>=1 && brs <4 && kol >= 1 && kol <4)
        return 1;
    if (brs>=1 && brs <4 && kol >= 5 && kol <8)
        return 2;
    if (brs>=5 && brs <8 && kol >= 1 && kol <4)
        return 3;
    if (brs>=5 && brs <8 && kol >= 1 && kol <4)
        return 4;
    else return 0;
}
```

```

public boolean cekhyper (int nokotakhyper, int nilai)
//mengecek apakah dalam satu kotak 3x3 hyper terdapat angka yang sama dengan nilai yang akan diassign
{
    switch (nokotakhyper) {
        case 0 : return true;
        case 1 : return this.Matriks[1][1] != nilai && this.Matriks[1][2] != nilai && this.Matriks[1][3] != nilai &&
this.Matriks[2][1] != nilai && this.Matriks[2][2] != nilai && this.Matriks[2][3] != nilai &&
this.Matriks[3][1] != nilai && this.Matriks[3][2] != nilai && this.Matriks[3][3] != nilai;
        case 2 : return this.Matriks[1][5] != nilai && this.Matriks[1][6] != nilai && this.Matriks[1][7] != nilai &&
this.Matriks[2][5] != nilai && this.Matriks[2][6] != nilai && this.Matriks[2][7] != nilai &&
this.Matriks[3][5] != nilai && this.Matriks[3][6] != nilai && this.Matriks[3][7] != nilai;
        case 3 : return this.Matriks[5][1] != nilai && this.Matriks[5][2] != nilai && this.Matriks[5][3] != nilai &&
this.Matriks[6][1] != nilai && this.Matriks[6][2] != nilai && this.Matriks[6][3] != nilai &&
this.Matriks[7][1] != nilai && this.Matriks[7][2] != nilai && this.Matriks[7][3] != nilai;
        case 4: return this.Matriks[5][5] != nilai && this.Matriks[5][6] != nilai && this.Matriks[5][7] != nilai &&
this.Matriks[6][5] != nilai && this.Matriks[6][7] != nilai && this.Matriks[6][7] != nilai &&
this.Matriks[7][5] != nilai && this.Matriks[7][7] != nilai && this.Matriks[7][7] != nilai;
        default : return true;
    }
}

public boolean ceksemua(int brs, int kol, int nilai)
//mengecek apakah suatu nilai mungkin dimasukkan ke dalam matriks ke brs dan kol
{
    return (cekbrs (brs, nilai) && cekkol(kol, nilai) && cekbox(nokotak(brs-brs%3, kol-kol%3), nilai) &&
cekhyper(nokotakhyper(brs, kol), nilai));
}

public boolean solve()
{
    memory m = new memory(0,0);
    int brs, kol, nilai;

    if (lisfilled(m))
        return true;

    for (nilai=1; nilai <= 9; nilai++)
    {
        count++;
        if (ceksemua(m.brs, m.kol, nilai))
        {
            Matriks[m.brs][m.kol] = nilai;
            if (solve())
                return true;
            Matriks[m.brs][m.kol] = 0;
        }
    }
    return false;
}
}

```

Main program

```
package hypersudoku;

public class Hypersudoku {

    public static void main(String[] args) {
        // TODO code application logic here

        matriks solver = new matriks();

        solver.isimatriks();
        System.out.println("sudoku awal : ");
        solver.printmatriks();

        long start = System.nanoTime();
        if (solver.solve())
        {
            System.out.println("Jawaban :");
            solver.printmatriks();
            System.out.println("Total Assignment : " + solver.count);
        }
        else
        {
            System.out.println("Tidak ada solusi");
        }

        long end = System.nanoTime();
        long duration = (end-start)/1000000;

        System.out.println("Waktu Eksekusi " + duration + " ms");
    }
}
```

Berikut adalah contoh input dan output dari program *sudoku solver*

```
E:\documents\Semester 4\Strategi Algoritma\hypersudoku\dist>java -jar hypersudoku.jar
```

```
sudoku awal :
```

```
+-----+
|0|0|0|3|5|0|0|0|7|
|0|0|0|8|0|0|0|9|0|
|0|2|0|0|0|0|0|0|0|
+-----+
```

```
|0|0|9|1|7|0|0|0|0|
|0|7|0|0|0|0|0|4|0|
|0|0|0|0|3|4|9|0|0|
+-----+
```

```
|0|0|0|0|0|0|0|8|0|
|0|5|0|0|0|6|0|0|0|
|6|0|0|0|9|8|0|0|0|
+-----+
```

```
Jawaban :
```

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

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

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

```
Total Assignment :15792
```

```
Waktu Eksekusi 25 ms
```

sudoku - Notepad

File Edit Format View Help

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

```
E:\documents\Semester 4\Strategi Algoritma\hypersudoku\dist>java -jar hypersudoku.jar
```

```
sudoku awal :
```

```
+-----+
|0|0|6|0|0|0|0|0|0|
|0|0|0|7|0|0|1|0|0|
|0|0|9|5|0|6|2|0|0|
+-----+
```

```
|0|0|0|0|0|0|0|3|0|
|5|4|3|0|0|0|9|1|7|
|0|7|0|0|0|0|0|0|0|
+-----+
```

```
|0|0|5|6|0|8|3|0|0|
|0|0|8|0|0|1|0|0|0|
|0|0|0|0|0|0|4|0|0|
+-----+
```

```
Jawaban :
```

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

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

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

```
Total Assignment :17539
```

```
Waktu Eksekusi 29 ms
```

sudoku - Notepad

File Edit Format View Help

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

```
E:\documents\Semester 4\Strategi Algoritma\hypersudoku\dist>java -jar hypersudoku.jar
```

```
sudoku awal :
```

```
+-----+
|0|4|0|0|0|0|7|0| |
|7|8|0|0|1|0|0|6|9|
|0|0|0|0|0|0|0|0|
+-----+
|0|0|0|4|0|1|0|0| |
|0|5|0|0|3|0|0|4|0|
|0|0|0|7|0|6|0|0|0|
+-----+
|0|0|0|0|0|0|0|0| |
|1|6|0|0|9|0|0|3|8|
|0|7|0|0|0|0|0|2|0|
+-----+
```

```
Jawaban :
```

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

```
Total Assignment :139500
```

```
Waktu Eksekusi 73 ms
```

sudoku - Notepad

File Edit Format View Help

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

```
E:\documents\Semester 4\Strategi Algoritma\hypersudoku\dist>java -jar hypersudoku.jar
```

```
sudoku awal :
```

```
+-----+
|0|0|0|0|0|0|0|1|0|
|0|0|2|0|0|0|0|3|4|
|0|0|0|0|5|1|0|0|0|
+-----+
|0|0|0|0|0|6|5|0|0|
|0|7|0|3|0|0|0|8|0|
|0|0|3|0|0|0|0|0|0|
+-----+
|0|0|0|0|8|0|0|0|0|
|5|8|0|0|0|0|9|0|0|
|6|9|0|0|0|0|0|0|0|
+-----+
```

```
Jawaban :
```

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

```
Total Assignment :440016
```

```
Waktu Eksekusi 84 ms
```

sudoku - Notepad

File Edit Format View Help

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



```
E:\documents\Semester 4\Strategi Algoritma\hypersudoku\dist>java -jar hypersudoku.jar
sudoku awal :
+-----+
|5|3|0|0|7|0|0|0|0|
|6|0|0|1|9|5|0|0|0|
|0|9|8|0|0|0|0|6|0|
+-----+
|8|0|0|0|6|0|0|0|3|
|4|0|0|8|0|3|0|0|1|
|7|0|0|0|2|0|0|0|6|
+-----+
|0|6|0|0|0|0|2|8|0|
|0|0|0|4|1|9|0|0|5|
|0|0|0|0|8|0|0|7|9|
+-----+
Tidak ada solusi
Waktu Eksekusi 6 ms
```

sudoku - Notepad

File	Edit	Format	View	Help
------	------	--------	------	------

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

Poin	Ya	Tidak
1. Program berhasil dikompilasi	√	
2. Program berhasil <i>running</i>	√	
3. Program dapat membaca file input dan menuliskan luaran.	√	
4. Luaran sudah benar untuk semua data uji (solusi <i>hyper-sudoku</i> benar)	√	