

# Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

Who? Michael Adrian  
2013730039  
michaeladrian39@gmail.com

From? Program Studi Teknik Informatika  
Fakultas Teknologi Informasi dan Sains  
Universitas Katolik Parahyangan

When? 20 Desember 2017

2017-12-19

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

Who? Michael Adrian  
2013730039  
michaeladrian39@gmail.com

From? Program Studi Teknik Informatika  
Fakultas Teknologi Informasi dan Sains  
Universitas Katolik Parahyangan

When? 20 Desember 2017

# Calcudoku

- Salah satu jenis permainan teka-teki aritmatika dan logika
- Dikenal juga sebagai KenKen, KenDoku, atau Mathdoku

2017-12-19

## Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Pendahuluan

- Calcudoku

- Calcudoku

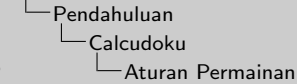
- Salah satu jenis permainan teka-teki aritmatika dan logika
- Dikenal juga sebagai KenKen, KenDoku, atau Mathdoku

## Aturan Permainan

- Pemain diberikan sebuah *grid* dengan ukuran  $n \times n$
- $n$  biasanya antara 3 sampai dengan 9
- *Grid* ini harus diisi dengan angka 1 sampai dengan  $n$
- Dalam setiap baris setiap angka hanya muncul sekali
- Dalam setiap kolom setiap angka hanya muncul sekali
- *Grid* dibagi ke dalam *cage*
- *Cage* adalah sekelompok sel yang dibatasi oleh garis yang lebih tebal daripada garis pembatas antar sel dengan angka tujuan dan operator yang telah ditentukan
- Angka-angka dalam setiap *cage* harus mencapai angka tujuan jika dihitung menggunakan operator yang telah ditentukan
- Angka tujuan dan operasi yang telah ditentukan ditulis di sudut kiri atas *cage*

2017-12-19

# Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku



Seperti dalam Sudoku, dalam teka-teki ini, pemain diberikan sebuah *grid* dengan ukuran  $n \times n$ , dengan  $n$  biasanya antara 3 sampai dengan 9. *Grid* ini harus diisi dengan angka 1 sampai dengan  $n$  sehingga dalam setiap baris setiap angka hanya muncul sekali, dalam setiap kolom setiap angka hanya muncul sekali. Perbedaannya dengan Sudoku adalah, Calcudoku dibagi ke dalam *cage* (sekelompok sel yang dibatasi oleh garis yang lebih tebal daripada garis pembatas antar sel dengan angka tujuan dan operator yang telah ditentukan), dan angka-angka dalam setiap *cage* harus mencapai angka tujuan jika dihitung menggunakan operator yang telah ditentukan. Angka tujuan dan operasi yang telah ditentukan ditulis di sudut kiri atas *cage*.

- Pemain diberikan sebuah grid dengan ukuran  $n \times n$
- $n$  bilangan antara 3 sampai dengan 9
- Isi grid ini harus diisi dengan angka 1 sampai dengan  $n$
- Dalam setiap baris setiap angka hanya muncul sekali
- Dalam setiap kolom setiap angka hanya muncul sekali
- Grid dibagi ke dalam cage
- Cage adalah sekelompok sel yang dibatasi oleh garis yang lebih tebal daripada garis pembatas antar sel dengan angka tujuan dan operator yang telah ditentukan
- Angka-angka dalam setiap cage harus mencapai angka tujuan jika dihitung menggunakan operator yang telah ditentukan
- Angka tujuan dan operasi yang telah ditentukan ditulis di sudut kiri atas cage

# Operator-Operator Matematika

- Ada 5 kemungkinan operator:
  - $+$  (penjumlahan)
  - $-$  (pengurangan)
  - $\times$  (perkalian)
  - $\div$  (pembagian)
  - $=$  (sama dengan)
- Jika operasi matematika yang ditentukan adalah pengurangan atau pembagian, maka ukuran cage harus berukuran dua sel

Ada lima kemungkinan operator:

1.  $+$ , sebuah operator  $n$ -ary yang menandakan penjumlahan.
2.  $-$ , sebuah operator biner yang menandakan pengurangan.
3.  $\times$ , sebuah operator  $n$ -ary yang menandakan perkalian.
4.  $\div$  sebuah operator biner yang menandakan pembagian.
5.  $=$ , (simbol ini biasanya dihilangkan), sebuah operator uner yang menandakan persamaan.

Jika operasi matematika yang ditentukan adalah pengurangan atau pembagian, maka ukuran *cage* harus berukuran dua sel. Pada beberapa versi dari teka-teki ini, hanya angka tujuan yang diberikan, dan pemain harus menebak operator dari setiap *cage* untuk menyelesaikan teka-tekinya.

- Ada 5 kemungkinan operator:
  - + (penjumlahan)
  - - (pengurangan)
  - × (perkalian)
  - ÷ (pembagian)
  - = (sama dengan)
- Jika operasi matematika yang ditentukan adalah pengurangan atau pembagian, maka ukuran cage harus berukuran dua sel

# Contoh Permainan

3	8 +	3 -	
7 +			
	8 +	8 +	
			1

Figure 1: Contoh permainan teka-teki dengan ukuran *grid* 4 x 4 yang belum diselesaikan.

2017-12-19

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└─ Pendahuluan

└─ Calcudoku

└─ Contoh Permainan

Contoh Permainan

3	8 +	3 -	
7 +			
	8 +	8 +	
			1

Figure 1: Contoh permainan teka-teki dengan ukuran *grid* 4 x 4 yang belum diselesaikan.

## Contoh Solusi

<sup>3</sup> 3	<sup>8+</sup> 2	<sup>3-</sup> 1	4
<sup>7+</sup> 1	4	2	3
4	<sup>8+</sup> 1	<sup>8+</sup> 3	2
2	3	4	<sup>1</sup> 1

Figure 2: Solusi untuk permainan teka-teki Calcudoku yang diberikan pada Gambar 1.

2017-12-19

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└─ Pendahuluan

└─ Calcudoku

└─ Contoh Solusi

Contoh Solusi

<sup>1</sup> 3	<sup>1+</sup> 2	<sup>1+</sup> 1	4
<sup>1+</sup> 1	4	2	3
4	<sup>1+</sup> 1	<sup>1+</sup> 3	2
2	3	4	1

Figure 2: Solusi untuk permainan teka-teki Calcudoku yang diberikan pada Gambar 1.

## Algoritma *Backtracking*

- Sebuah algoritma umum yang mencari solusi dengan mencoba salah satu dari beberapa pilihan, jika pilihan yang dipilih ternyata salah, komputasi dimulai lagi pada titik pilihan dan mencoba pilihan lainnya
- Untuk bisa melacak kembali langkah-langkah yang telah dipilih, maka algoritma harus secara eksplisit menyimpan jejak dari setiap langkah yang sudah pernah dipilih, atau menggunakan rekursi (*recursion*)
- Rekursi dipilih karena jauh lebih mudah daripada harus menyimpan jejak setiap langkah yang pernah dipilih
- Hal ini menyebabkan algoritma ini biasanya berbasis DFS (*Depth First Search*)

2017-12-19

## Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Pendahuluan

## – Algoritma *Backtracking*

### – Algoritma *Backtracking*

Algoritma *backtracking* adalah sebuah algoritma umum yang mencari solusi dengan mencoba salah satu dari beberapa pilihan, jika pilihan yang dipilih ternyata salah, komputasi dimulai lagi pada titik pilihan dan mencoba pilihan lainnya. Untuk bisa melacak kembali langkah-langkah yang telah dipilih, maka algoritma harus secara eksplisit menyimpan jejak dari setiap langkah yang sudah pernah dipilih, atau menggunakan rekursi (*recursion*). Rekursi dipilih karena jauh lebih mudah daripada harus menyimpan jejak setiap langkah yang pernah dipilih. Hal ini menyebabkan algoritma ini biasanya berbasis DFS (*Depth First Search*).

- Sebuah algoritma umum yang mencari solusi dengan mencoba salah satu dari beberapa pilihan, jika pilihan yang dipilih ternyata salah, komputasi dimulai lagi pada titik pilihan dan mencoba pilihan lainnya
- Untuk bisa melacak kembali langkah-langkah yang telah dipilih, maka algoritma harus secara eksplisit menyimpan jejak dari setiap langkah yang sudah pernah dipilih, atau menggunakan rekursi (recursion)
- Rekursi dipilih karena jauh lebih mudah daripada harus menyimpan jejak setiap langkah yang pernah dipilih
- Hal ini menyebabkan algoritma ini biasanya berbasis DFS (Depth First Search)

## Algoritma *Hybrid Genetic*

- Dalam kasus ini, algoritma *hybrid genetic* adalah gabungan dari algoritma *rule based* dan algoritma genetik
- Algoritma *rule based* akan dijalankan sampai pada titik dimana algoritma tidak bisa menyelesaikan permainan teka-teki Calcudoku
- Jika algoritma sudah tidak bisa menyelesaikan permainan, maka algoritma genetik akan mulai dijalankan

2017-12-19

## Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Pendahuluan

### —Algoritma *Hybrid Genetic*

### —Algoritma *Hybrid Genetic*

Dalam kasus ini, algoritma ini gabungan dari algoritma *rule based* dan algoritma genetik. Algoritma *rule based* akan dijalankan sampai pada titik dimana algoritma tidak bisa menyelesaikan permainan teka-teki Calcudoku. Jika algoritma sudah tidak bisa menyelesaikan permainan, maka algoritma genetik akan mulai dijalankan.

- Dalam kasus ini, algoritma *hybrid genetic* adalah gabungan dari algoritma *rule based* dan algoritma *genetik*.
- Algoritma *rule based* akan dijalankan sampai pada titik dimana algoritma tidak bisa menyelesaikan permainan teka-teki *Caludoku*.
- Jika algoritma sudah tidak bisa menyelesaikan permainan, maka algoritma *genetik* akan mulai dijalankan.



## Algoritma *Rule Based*

- Sebuah algoritma berbasis aturan logika untuk menyelesaikan teka-teki Sudoku dan variasinya, termasuk Calcudoku
- Beberapa aturan logika yang digunakan dalam algoritma ini adalah:
  - *Single square rule*
  - *Naked subset rule*
  - *Hidden single rule*
  - *Killer combination*

2017-12-19

## Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Pendahuluan

### —Algoritma *Hybrid Genetic*

- Algoritma *Rule Based*

Algoritma *rule based* adalah sebuah algoritma berbasis aturan logika untuk menyelesaikan teka-teki Sudoku dan variasinya, termasuk Calcudoku. Menurut Johanna, Lukas, dan Saputra, beberapa aturan logika yang digunakan dalam algoritma ini adalah *single square rule*, *naked subset rule*, *hidden single rule*, *evil twin rule*, *killer combination*, dan *X-wing* [2].

- Sebuah algoritma berbasis aturan logika untuk menyelesaikan teka-teki Sudoku dan variasinya, termasuk Calcdoku
- Beberapa aturan logika yang digunakan dalam algoritma ini adalah:
  - *Single square rule*
  - *Naked subset rule*
  - *Hidden single rule*
  - *Killer combination*

# Algoritma Genetik

- Salah satu teknik heuristik *Generate and Test* yang terinspirasi oleh sistem seleksi alam
- Perpaduan dari bidang biologi dan ilmu komputer.
- Algoritma ini memanipulasi informasi, biasanya disebut sebagai kromosom.
- Kromosom ini meng-*encode* kemungkinan jawaban untuk sebuah masalah yang diberikan.
- Kromosom dievaluasi dan diberi *fitness value*.
- *Child chromosome*) diproduksi dengan menggabungkan dua (atau lebih) *parent chromosome*.

2017-12-19

## Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

## – Pendahuluan

### – Algoritma *Hybrid Genetic*

- Algoritma Genetik

- Salah satu teknik heuristik *Generate and Test* yang terinspirasi oleh sistem seleksi alam.
- Perpaduan dari bidang biologi dan ilmu komputer.
- Algoritma ini memanipulasi informasi, biasanya disebut sebagai kromosom.
- Kromosom ini meng-encode kemungkinan jawaban untuk sebuah masalah yang diberikan.
- Kromosom dievaluasi dan diberi *fitness value*.
- Child chromosome* diproduksi dengan menggabungkan dua (atau lebih) *parent chromosome*.

Algoritma genetika adalah salah satu teknik heuristik *Generate and Test* yang terinspirasi oleh sistem seleksi alam. Algoritma ini adalah perpaduan dari bidang biologi dan ilmu komputer. Algoritma ini memanipulasi informasi, biasanya disebut sebagai kromosom. Kromosom ini meng-*encode* kemungkinan jawaban untuk sebuah masalah yang diberikan. Kromosom dievaluasi dan diberi *fitness value* berdasarkan seberapa baik kromosom dalam menyelesaikan masalah yang diberikan berdasarkan kriteria yang ditentukan oleh pembuat program. Nilai kelayakan ini digunakan sebagai probabilitas keberlangsungan hidup kromosom dalam satu siklus reproduksi. Kromosom baru (kromosom anak, *child chromosome*) diproduksi dengan menggabungkan dua (atau lebih) kromosom orang tua (*parent chromosome*). Proses ini dirancang untuk menghasilkan kromosom-kromosom keturunan yang lebih layak, kromosom-kromosom ini meng-*encode* jawaban yang lebih baik, sampai solusi yang baik dan yang bisa diterima ditemukan.

## Tujuan

- Tujuan dari penelitian ini adalah:
  - 1 Membuat perangkat lunak permainan teka-teki Calcudoku.
  - 2 Membandingkan performansi algoritma *backtracking* dengan algoritma *hybrid genetic*.
- Ruang lingkup dari penelitian ini dibatasi oleh batasan-batasan masalah sebagai berikut:
  - 1 Ukuran *grid* untuk permainan teka-teki Calcudoku adalah antara  $4 \times 4$  sampai dengan  $8 \times 8$ .
  - 2 Pada algoritma *rule based*, aturan-aturan logika yang digunakan dibatasi hanya pada aturan *single square*, *naked single*, *naked double*, *hidden single*, dan *killer combination*.

2017-12-19

## Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Pendahuluan

### —Algoritma *Hybrid Genetic*

– Tujuan

- 1 Tujuan dari penelitian ini adalah:
  - 1 Membuat perangkat lunak permainan teka-teki Caludoku
  - 2 Membandingkan performansi algoritma backtracking dengan algoritma hybrid genetic
- 2 Ruang lingkup dari penelitian ini dibatasi oleh batasan-batasan masalah sebagai berikut:
  - 1 Ukuran grid untuk permainan teka-teki Caludoku adalah  $4 \times 4$  sampai dengan  $8 \times 8$ .
  - 2 Pada algoritma rule based, aturan-aturan logika yang digunakan dibatasi hanya pada aturan single, naked single, naked double, hidden single, dan killer combination

Tujuan dari penelitian ini adalah:

1. Membuat perangkat lunak permainan teka-teki Calcudoku yang menerima input berupa soal teka-teki dan mampu menyelesaikan soal teka-teki tersebut menggunakan algoritma *backtracking* dan *hybrid genetic*.
2. Membandingkan performansi algoritma *backtracking* dengan algoritma *hybrid genetic* dalam hal kecepatan dan kesuksesan dalam menyelesaikan Calcudoku.

Ruang lingkup dari penelitian ini dibatasi oleh batasan-batasan masalah sebagai berikut:

1. Ukuran *grid* untuk permainan teka-teki Calcudoku adalah antara  $4 \times 4$  sampai dengan  $8 \times 8$ .
2. Pada algoritma *rule based*, yang merupakan bagian dari algoritma *hybrid genetic*, aturan-aturan logika yang digunakan dibatasi hanya pada aturan *single square*, *naked single*, *naked double*, *hidden single*, dan *killer combination*.

# Analisis Perangkat Lunak

- Perangkat lunak ini akan menerima masukan dalam bentuk *file* yang berisi:
  - 1 Ukuran *grid*
  - 2 Jumlah *cage*
  - 3 Matriks *cage assignment*
  - 4 Matriks *cage objectives*
- Perangkat lunak ini akan menghasilkan keluaran berupa GUI permainan teka-teki Calcadoku berdasarkan isi *file* yang di-load oleh pengguna.
- Permainan ini dapat diselesaikan oleh pengguna, atau menggunakan salah satu dari dua *solver* yang disediakan. Kedua *solver* tersebut yaitu:
  - 1 Algoritma *backtracking*, dan
  - 2 Algoritma *hybrid genetic*.

2017-12-19

## Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcadoku

### Analisis

### Analisis Perangkat Lunak

Perangkat lunak ini akan menerima masukan dalam bentuk *file* yang berisi:

1. Ukuran *grid*.
2. Jumlah *cage*.
3. Matriks *cage assignment*, yang merepresentasikan posisi dari setiap *cage* dalam *grid*.
4. Matriks *cage objectives*, yang berisikan angka tujuan dan operasi matematika yang telah ditentukan untuk setiap *cage*.

Perangkat lunak ini akan menghasilkan keluaran berupa antarmuka grafis permainan teka-teki Calcadoku berdasarkan isi *file* yang di-load oleh pengguna. Permainan ini dapat diselesaikan oleh pengguna dengan usahanya sendiri, atau menggunakan salah satu dari dua *solver* yang disediakan. Kedua *solver* tersebut yaitu:

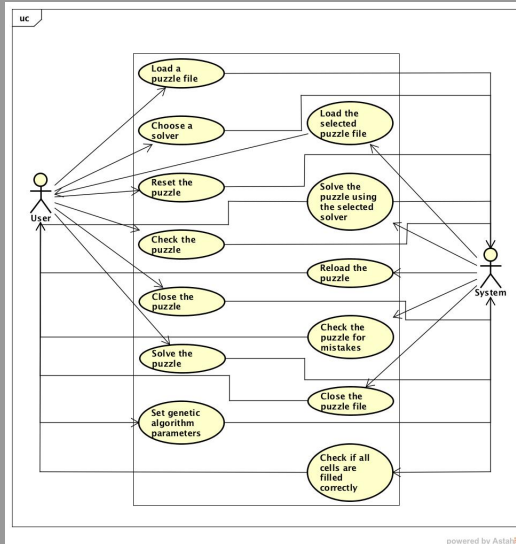
1. Algoritma *backtracking*, dan
2. Algoritma *hybrid genetic*.

Analisis Perangkat Lunak

- Perangkat lunak ini akan menerima masukan dalam bentuk *file* yang berisi:
  - 1 Ukuran *grid*
  - 2 Jumlah *cage*
  - 3 Matriks *cage assignment*
  - 4 Matriks *cage objectives*
- Perangkat lunak ini akan menghasilkan keluaran berupa GUI permainan teka-teki Calcadoku berdasarkan isi *file* yang di-load oleh pengguna.
- Permainan ini dapat diselesaikan oleh pengguna, atau menggunakan salah satu dari dua *solver* yang disediakan. Kedua *solver* tersebut yaitu:
  - 1 Algoritma *backtracking*, dan
  - 2 Algoritma *hybrid genetic*.

# Diagram Use Case

- Diagram *use case* berdasarkan skenario-skenario tersebut dapat dilihat pada Gambar 3.



Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

2017-12-19

Analisis

Diagram Use Case

Diagram Use Case  
Diagram use case berdasarkan skenario-skenario tersebut dapat dilihat pada Gambar 3.



Diagram *use case* adalah diagram yang menggambarkan interaksi antara sistem (perangkat lunak) dengan pengguna.

Skenario-skenario yang dapat dilakukan oleh pengguna adalah:

1. Membuka file masukan untuk memulai permainan.
2. Memilih salah satu dari dua *solver* yang disediakan untuk menyelesaikan permainan berdasarkan *file* yang sudah di-load.
3. Me-reset permainan untuk mengulang permainan berdasarkan *file* masukan yang sudah di-load dari awal.
4. Meminta perangkat lunak untuk memeriksa permainan jika ada masukan yang salah di dalam *grid*.
5. Menutup *file* masukan untuk mengakhiri permainan.
6. Menyelesaikan permainan dengan usahanya sendiri.
7. Mengatur nilai dari parameter-parameter untuk algoritma genetik.

Diagram *use case* berdasarkan skenario-skenario tersebut dapat dilihat pada Gambar 3.

# Perancangan Masukan

- Masukan untuk perangkat lunak Calcludoku ini berupa sebuah *file* teks, seperti yang ditunjukkan pada Gambar 4.

```
4
9
1 2 3 3
1 4 4 5
6 7 7 5
8 8 9 9
7+
2=
2-
3-
2/
1=
6*
3+
7+
```

Figure 4: Contoh *file* masukan.

2017-12-19

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcludoku

└ Perancangan

└ Perancangan Masukan

Masukan untuk perangkat lunak permainan teka-teki Calcludoku ini berupa sebuah *file* teks, seperti yang ditunjukkan pada Gambar 4.

Adapun rincian dari *file* teks masukan tersebut adalah sebagai berikut:

1. Baris pertama berisi ukuran *grid* dan banyaknya *cage* dari teka-teki Calcludoku tersebut. Angka pertama adalah ukuran *grid*, dan angka kedua adalah banyaknya *cage*.
2. Baris kedua sampai ke baris ke-2 + ( $n - 1$ ), dengan  $n$  adalah ukuran *grid*, berisi matriks *cage assignment*. Matriks ini merepresentasikan posisi dari setiap *cage* dalam *grid*. Setiap *cage* direpresentasikan dengan angka yang berbeda. Setiap *cage* dapat mempunyai ukuran (jumlah sel yang terdapat dalam *cage*) yang bervariasi. Setiap sel dalam sebuah *cage* harus berhubungan secara horizontal atau vertikal dengan sel lain dalam *cage* yang sama.
3. Baris ke-2 +  $n$  dan seterusnya berisi *cage objectives* untuk setiap *cage*. *Cage objectives* berisikan angka tujuan dan operasi matematika yang telah ditentukan. Angka-angka dalam sebuah *cage* harus mencapai angka tujuan jika dihitung menggunakan operasi matematika yang telah ditentukan.

Perancangan Masukan

- Masukan untuk perangkat lunak Calcludoku ini berupa sebuah *file* teks, seperti yang ditunjukkan pada Gambar 4.

```
4
9
1 2 3 3
1 4 4 5
6 7 7 5
8 8 9 9
7+
2=
2-
3-
2/
1=
6*
3+
7+
```

Figure 4: Contoh *file* masukan.

# Perancangan Keluaran

- Keluaran untuk perangkat lunak Calcudoku ini berupa sebuah matriks yang berisi solusi dari teka-teki Calcudoku yang sudah diselesaikan oleh program, seperti dapat dilihat pada Gambar 5.

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

Figure 5: Contoh keluaran.

2017-12-19

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Perancangan

└ Perancangan Keluaran

Keluaran untuk perangkat lunak permainan teka-teki Calcudoku ini berupa sebuah matriks yang berisi solusi dari teka-teki Calcudoku yang sudah diselesaikan oleh program, seperti dapat dilihat pada Gambar 5.

Perancangan Keluaran

- Keluaran untuk perangkat lunak Calcudoku ini berupa sebuah matriks yang berisi solusi dari teka-teki Calcudoku yang sudah diselesaikan oleh program, seperti dapat dilihat pada Gambar 5.

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

Figure 5: Contoh keluaran.

# Diagram Kelas

- Diagram kelas untuk perangkat lunak ini dapat dilihat pada Gambar 6.

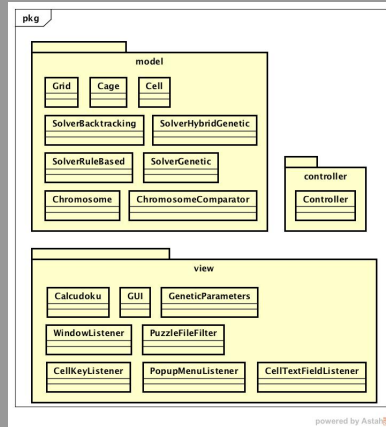


Figure 6: Diagram kelas untuk perangkat lunak Calcudoku.

2017-12-19

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Perancangan

└ Diagram Kelas

Diagram Kelas  
■ Diagram kelas untuk perangkat lunak ini dapat dilihat pada Gambar 6.



Figure 6: Diagram kelas untuk perangkat lunak Calcudoku.

Diagram kelas untuk perangkat lunak ini dapat dilihat pada Gambar 6.



## Kode Program

2017-12-19

## Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

- Kode Program

└─ Kode Program

# Hasil Pengujian - Algoritma *Backtracking*

- Pengujian algoritma *backtracking* dilakukan pada permainan dengan *grid* yang berukuran  $4 \times 4$  sampai dengan  $8 \times 8$ .
- Hasil pengujian algoritma *backtracking* dapat dilihat pada Tabel 1.

Table 1: Hasil pengujian algoritma *backtracking* untuk Calcudoku

Ukuran <i>Grid</i>	Rata-Rata Tingkat Keberhasilan	Rata-Rata Kecepatan
$4 \times 4$	100%	0.067 detik
$5 \times 5$	100%	0.701 detik
$6 \times 6$	100%	13.84 detik
$7 \times 7$	100%	482.653 detik
$8 \times 8$	100%	2134.655 detik

2017-12-19

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Implementasi dan Pengujian

└ Hasil Pengujian - Algoritma *Backtracking*

Hasil Pengujian - Algoritma *Backtracking*

- Pengujian algoritma *backtracking* dilakukan pada permainan dengan *grid* yang berukuran  $4 \times 4$  sampai dengan  $8 \times 8$ .
- Hasil pengujian algoritma *backtracking* dapat dilihat pada Tabel 1.

Table 1: Hasil pengujian algoritma *backtracking* untuk Calcudoku

Ukuran Grid	Rata-Rata Tingkat Keberhasilan	Rata-Rata Kecepatan
$4 \times 4$	100%	0.067 detik
$5 \times 5$	100%	0.701 detik
$6 \times 6$	100%	13.84 detik
$7 \times 7$	100%	482.653 detik
$8 \times 8$	100%	2134.655 detik

Pengujian algoritma *backtracking* dilakukan pada permainan dengan *grid* berukuran  $4 \times 4$  sampai dengan  $8 \times 8$ . Hasil pengujian algoritma *backtracking* dapat dilihat pada Tabel 1.

## Hasil Pengujian - Algoritma *Hybrid Genetic*

- Pengujian algoritma *hybrid genetic* dilakukan pada *grid* dengan ukuran  $4 \times 4$  sampai dengan  $8 \times 8$ .
- Dilakukan 16 skenario pengujian.
- Nilai dari parameter-parameter untuk algoritma genetik berbeda-beda untuk setiap skenario.
- Algoritma *hybrid genetic* gagal dalam menyelesaikan permainan dengan *grid* yang berukuran  $6 \times 6$  ke atas.

2017-12-19

## Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

## Implementasi dan Pengujian

└ Hasil Pengujian - Algoritma *Hybrid Genetic*

Pengujian algoritma *hybrid genetic* dilakukan pada *grid* dengan ukuran  $4 \times 4$  sampai dengan  $8 \times 8$ . Dilakukan 16 skenario pengujian. Nilai dari parameter-parameter untuk algoritma genetik berbeda-beda untuk setiap skenario. Algoritma *hybrid genetic* gagal dalam menyelesaikan permainan dengan *grid* yang berukuran  $6 \times 6$  ke atas.

# Hasil Pengujian - Algoritma *Hybrid Genetic*

- Daftar nilai dari parameter-parameter untuk algoritma genetik untuk setiap skenario dapat dilihat pada Tabel 2.

Table 2: Nilai untuk parameter-parameter algoritma genetik untuk setiap percobaan yang dilakukan

Skenario	Populasi	Generasi	<i>Elitism</i>	Mutasi	Kawin Silang
1	1000	100	10%	10%	80%
2	1000	100	5%	10%	85%
3	1000	100	10%	5%	85%
4	1000	100	5%	5%	90%
5	100	100	10%	10%	80%
6	100	100	5%	10%	85%
7	100	100	10%	5%	85%
8	100	100	5%	5%	90%
9	1000	10	10%	10%	80%
10	1000	10	5%	10%	85%
11	1000	10	10%	5%	85%
12	1000	10	5%	5%	90%
13	100	10	10%	10%	80%
14	100	10	5%	10%	85%
15	100	10	10%	5%	85%
16	100	10	5%	5%	90%

2017-12-19

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Implementasi dan Pengujian

└ Hasil Pengujian - Algoritma *Hybrid Genetic*

Daftar nilai dari parameter-parameter untuk algoritma genetik untuk setiap skenario dapat dilihat pada Tabel 2.

Hasil Pengujian - Algoritma *Hybrid Genetic*

- Daftar nilai dari parameter-parameter untuk algoritma genetik untuk setiap skenario dapat dilihat pada Tabel 2.

Table 2: Nilai untuk parameter-parameter algoritma genetik untuk setiap percobaan yang dilakukan

Skenario	Populasi	Generasi	<i>Elitism</i>	Mutasi	Kawin Silang
1	1000	100	10%	10%	80%
2	1000	100	5%	10%	85%
3	1000	100	10%	5%	85%
4	1000	100	5%	5%	90%
5	100	100	10%	10%	80%
6	100	100	5%	10%	85%
7	100	100	10%	5%	85%
8	100	100	5%	5%	90%
9	1000	10	10%	10%	80%
10	1000	10	5%	10%	85%
11	1000	10	10%	5%	85%
12	1000	10	5%	5%	90%
13	100	10	10%	10%	80%
14	100	10	5%	10%	85%
15	100	10	10%	5%	85%
16	100	10	5%	5%	90%

# Hasil Pengujian - Algoritma *Hybrid Genetic*

- Hasil pengujian algoritma *hybrid genetic* untuk Calcudoku dengan *grid* berukuran  $4 \times 4$  dapat dilihat pada Tabel 4.

Table 3: Hasil pengujian algoritma *hybrid genetic* untuk Calcudoku dengan *grid* berukuran  $4 \times 4$

Skenario	Rata-Rata Tingkat Keberhasilan	Rata-Rata Kecepatan	Rata-Rata Jumlah Sel Diisi Algoritma <i>Rule Based</i>
1	100%	3.735 detik	3
2	100%	4.183 detik	3
3	100%	3.924 detik	3
4	100%	4.371 detik	3
5	56.41%	0.48 detik	5
6	56.41%	0.532 detik	5
7	56.41%	0.505 detik	5
8	56.41%	0.557 detik	5
9	33.333%	0.457 detik	7
10	33.333%	0.457 detik	7
11	33.333%	0.457 detik	7
12	33.333%	0.457 detik	7
13	23.077%	0.048 detik	9
14	23.077%	0.048 detik	9
15	23.077%	0.048 detik	9
16	23.077%	0.048 detik	9

## Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

### Implementasi dan Pengujian

#### Hasil Pengujian - Algoritma *Hybrid Genetic*

Hasil pengujian algoritma *hybrid genetic* untuk Calcudoku dengan *grid* berukuran  $4 \times 4$  dapat dilihat pada Tabel 4.

Hasil Pengujian - Algoritma *Hybrid Genetic*

- Hasil pengujian algoritma *hybrid genetic* untuk Calcudoku dengan *grid* berukuran  $4 \times 4$  dapat dilihat pada Tabel 4.

Table 3: Hasil pengujian algoritma *hybrid genetic* untuk Calcudoku dengan *grid* berukuran  $4 \times 4$

Skenario	Rata-Rata Tingkat Keberhasilan	Rata-Rata Kecepatan	Rata-Rata Jumlah Sel Diisi Algoritma <i>Rule Based</i>
1	100%	3.735 detik	3
2	100%	4.183 detik	3
3	100%	3.924 detik	3
4	100%	4.371 detik	3
5	56.41%	0.48 detik	5
6	56.41%	0.532 detik	5
7	56.41%	0.505 detik	5
8	56.41%	0.557 detik	5
9	33.333%	0.457 detik	7
10	33.333%	0.457 detik	7
11	33.333%	0.457 detik	7
12	33.333%	0.457 detik	7
13	23.077%	0.048 detik	9
14	23.077%	0.048 detik	9
15	23.077%	0.048 detik	9
16	23.077%	0.048 detik	9

# Hasil Pengujian - Algoritma *Hybrid Genetic*

- Hasil pengujian algoritma *hybrid genetic* untuk Calcudoku dengan *grid* berukuran  $5 \times 5$  dapat dilihat pada Tabel 4.

Table 4: Hasil pengujian algoritma *hybrid genetic* untuk Calcudoku dengan *grid* berukuran  $5 \times 5$

Skenario	Rata-Rata Tingkat Keberhasilan	Rata-Rata Kecepatan	Rata-Rata Jumlah Sel Diisi Algoritma <i>Rule Based</i>
1	42.308%	8.389 detik	9
2	42.308%	9.258 detik	9
3	42.308%	8.806 detik	9
4	42.308%	9.676 detik	9
5	19.231%	0.311 detik	14
6	19.231%	0.339 detik	14
7	19.231%	0.325 detik	14
8	19.231%	0.352 detik	14
9	15.385%	0.487 detik	15
10	15.385%	0.487 detik	15
11	15.385%	0.487 detik	15
12	15.385%	0.487 detik	15
13	15.385%	0.077 detik	15
14	15.385%	0.077 detik	15
15	15.385%	0.077 detik	15
16	15.385%	0.077 detik	15

## Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

### Implementasi dan Pengujian

#### Hasil Pengujian - Algoritma *Hybrid Genetic*

Hasil pengujian algoritma *hybrid genetic* untuk Calcudoku dengan *grid* berukuran  $5 \times 5$  dapat dilihat pada Tabel 4.

Hasil Pengujian - Algoritma *Hybrid Genetic*

- Hasil pengujian algoritma *hybrid genetic* untuk Calcudoku dengan *grid* berukuran  $5 \times 5$  dapat dilihat pada Tabel 4.

Table 4: Hasil pengujian algoritma *hybrid genetic* untuk Calcudoku dengan *grid* berukuran  $5 \times 5$

Skenario	Rata-Rata Tingkat Keberhasilan	Rata-Rata Kecepatan	Rata-Rata Jumlah Sel Diisi Algoritma <i>Rule Based</i>
1	42.308%	8.389 detik	9
2	42.308%	9.258 detik	9
3	42.308%	8.806 detik	9
4	42.308%	9.676 detik	9
5	19.231%	0.311 detik	14
6	19.231%	0.339 detik	14
7	19.231%	0.325 detik	14
8	19.231%	0.352 detik	14
9	15.385%	0.487 detik	15
10	15.385%	0.487 detik	15
11	15.385%	0.487 detik	15
12	15.385%	0.487 detik	15
13	15.385%	0.077 detik	15
14	15.385%	0.077 detik	15
15	15.385%	0.077 detik	15
16	15.385%	0.077 detik	15

# Demo Program

2017-12-19

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└─ Implementasi dan Pengujian

└─ Demo Program

Demo Program

## Simpulan

- Perangkat lunak permainan teka-teki Calcudoku dengan dua *solver*, yaitu *solver* dengan algoritma *backtracking* dan *solver* dengan algoritma *hybrid genetic*, berhasil dibuat.
- Algoritma *backtracking* dapat menyelesaikan semua permainan yang diujikan, tetapi pada ukuran *grid* yang besar, algoritma *backtracking* sangat lambat dalam menyelesaikan permainan.
- Ada kemungkinan algoritma *hybrid genetic* gagal dalam menyelesaikan permainan karena sifat acak dari algoritma *hybrid genetic* ini. Semakin besar ukuran *grid*, maka kemungkinan algoritma *hybrid genetic* gagal dalam menyelesaikan permainan semakin besar.

# Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

— Simpulan dan Saran

└─Simpulan

Perangkat lunak permainan teka-teki Calcludoku dengan dua *solver*, yaitu *solver* dengan algoritma *backtracking* dan *solver* dengan algoritma *hybrid genetic*, berhasil dibuat. Perangkat lunak ini menerima input berupa soal teka-teki dan mampu menyelesaikan soal teka-teki tersebut menggunakan algoritma *backtracking* dan *hybrid genetic*.

Algoritma *backtracking* dapat menyelesaikan semua permainan yang diujikan, tetapi pada ukuran *grid* yang besar, algoritma *backtracking* sangat lambat dalam menyelesaikan permainan.

Ada kemungkinan algoritma *hybrid genetic* gagal dalam menyelesaikan permainan karena sifat acak dari algoritma *hybrid genetic* ini. Semakin besar ukuran *grid*, maka kemungkinan algoritma *hybrid genetic* gagal dalam menyelesaikan permainan semakin besar.

- Perangkat lunak permainan teka-teki *Calculus* dengan dua solver, yaitu solver dengan algoritma *backtracking* dan solver dengan algoritma *hybrid genetic*, berhasil dibuat.
- Algoritma *backtracking* dapat menyelesaikan semua permainan yang diujikan, tetapi pada ukuran *grid* yang besar, algoritma *backtracking* sangat lambat dalam menyelesaikan permainan.
- Ada kemungkinan algoritma *hybrid genetic* gagal dalam menyelesaikan permainan karena sifat acak dari algoritma *hybrid genetic* ini. Semakin besar ukuran *grid*, maka kemungkinan algoritma *hybrid genetic* gagal dalam menyelesaikan permainan semakin besar.



## Simpulan

- Pada ukuran *grid* yang kecil, algoritma *hybrid genetic* cenderung menyelesaikan permainan lebih lambat daripada algoritma *backtracking*.
- Pada ukuran *grid* yang besar, algoritma *hybrid genetic* mungkin mampu menyelesaikan permainan lebih cepat daripada algoritma *backtracking*, tetapi hal ini tidak dapat dibuktikan karena algoritma *hybrid genetic* gagal dalam menyelesaikan permainan dengan ukuran *grid* yang besar.
- Banyaknya sel yang diisi dalam tahap algoritma *rule based* dan nilai dari parameter-parameter untuk algoritma genetik mempengaruhi kecepatan dan tingkat keberhasilan algoritma *hybrid genetic* dalam menyelesaikan permainan.

## Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

## Simpulan dan Saran

└─Simpulan

Pada ukuran *grid* yang kecil, algoritma *hybrid genetic* cenderung menyelesaikan permainan lebih lambat daripada algoritma *backtracking*. Tetapi, pada ukuran *grid* yang besar, algoritma *hybrid genetic* mungkin mampu menyelesaikan permainan lebih cepat daripada algoritma *backtracking*, tetapi hal ini tidak dapat dibuktikan karena algoritma *hybrid genetic* gagal dalam menyelesaikan permainan dengan ukuran *grid* yang besar.

Banyaknya sel yang diisi dalam tahap algoritma *rule based* dan nilai dari parameter-parameter untuk algoritma genetik mempengaruhi kecepatan dan keberhasilan algoritma *hybrid genetic* dalam menyelesaikan permainan.

Semakin banyak sel yang diisi dalam tahap algoritma *rule based*, semakin besar juga kemungkinan algoritma genetik untuk berhasil dalam menyelesaikan permainan dan semakin cepat algoritma genetik dalam menyelesaikan permainan.

Semakin besar populasi dalam sebuah generasi sampai ke titik tertentu, dan semakin banyak generasi sampai ke titik tertentu, maka semakin besar juga kemungkinan algoritma *hybrid genetic* berhasil dalam menyelesaikan permainan. Semakin besar tingkat *elitism* dan tingkat mutasi sampai ke titik tertentu, maka semakin cepat juga algoritma *hybrid genetic* dalam menyelesaikan permainan.

- Pada ukuran grid yang kecil, algoritma *hybrid genetic* cenderung menyelesaikan permainan lebih lambat daripada algoritma *backtracking*.
- Pada ukuran grid yang besar, algoritma *hybrid genetic* mungkin mampu menyelesaikan permainan lebih cepat daripada algoritma *backtracking*, tetapi hal ini tidak dapat dibuktikan karena algoritma *hybrid genetic* gagal dalam menyelesaikan permainan dengan ukuran grid yang besar.
- Banyaknya sel yang diisi dalam tahap algoritma *rule based* dan nilai dari parameter-parameter untuk algoritma genetik mempengaruhi kecepatan dan tingkat keberhasilan algoritma *hybrid genetic* dalam menyelesaikan permainan.

## Saran

- Memperbaiki GUI dari perangkat lunak ini agar petunjuk dapat ditampilkan sebagaimana mestinya.
- Menambah aturan-aturan logika untuk algoritma *rule based*, misalnya aturan *naked subset* untuk *cage* yang berukuran lebih besar dari 3 sel, aturan *hidden subset* untuk *cage* yang berukuran lebih besar dari 2 sel, aturan *killer combination* untuk *cage* yang berukuran lebih besar dari 2 sel.
- Memperbaiki algoritma genetik, misalnya proses pemberian nilai kelayakan untuk kromosom, proses pemilihan kromosom untuk kawin silang dan mutasi, proses *elitism*, proses kawin silang, dan proses mutasi.

2017-12-19

# Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

## Simpulan dan Saran

–Saran

Memperbaiki GUI dari perangkat lunak ini agar petunjuk, yaitu angka tujuan dan operasi matematika yang ditentukan untuk sebuah *cage*, dapat ditampilkan sebagaimana mestinya, yaitu pada di sudut kiri atas sel yang paling atas dan yang paling kiri dalam *cage* tersebut.

Menambah aturan-aturan logika untuk algoritma *rule based*, misalnya aturan *naked subset* untuk *cage* yang berukuran lebih besar dari 3 sel, aturan *hidden subset* untuk *cage* yang berukuran lebih besar dari 2 sel, aturan *killer combination* untuk *cage* yang berukuran lebih besar dari 2 sel, dan aturan *evil twin* untuk *cage* yang berukuran minimal 2 sel. Dengan menambah aturan-aturan logika untuk algoritma *rule based*. Diharapkan, dengan menambah aturan-aturan logika untuk algoritma *rule based*, maka tingkat kesuksesan algoritma *hybrid genetic* dalam menyelesaikan permainan Calcudoku dapat meningkat.

Memperbaiki algoritma genetika, misalnya proses pemberian nilai kelayakan untuk kromosom, proses pemilihan kromosom untuk kawin silang dan mutasi, proses *elitism*, proses kawin silang, dan proses mutasi, sehingga tingkat kesuksesan algoritma *hybrid genetic* dalam menyelesaikan permainan Calcudoku dapat meningkat.

- Memberipakai GUI dari perangkat lunak ini agar petunjuk dapat ditampilan sebagaimana mestinya.
- Menambah aturan-aturan logika untuk algoritma rule based, misalnya aturan naked subset untuk cage yang berukuran lebih besar dari 3 sel, aturan hidden subset untuk cage yang berukuran lebih besar dari 2 sel, aturan killer combination untuk cage yang berukuran lebih besar dari 2 sel.
- Memberipakai algoritma genetika, misalnya proses pemberian nilai kelayakan untuk kromosom, proses pemilihan kromosom untuk kawin silang dan mutasi, proses elitism, proses kawin silang, dan proses mutasi.

# Daftar Pustaka



Asanilta Fahda, *KenKen Puzzle Solver using Backtracking Algorithm*, Makalah IF2211 Strategi Algoritma - Semester II Tahun 2014/2015, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung 2015.



Olivia Johanna, Samuel Lukas, Kie Van Ivanky Saputra, *Solving and Modeling Ken-ken Puzzle by Using Hybrid Genetics Algorithm*, 1st International Conference on Engineering and Technology Development (ICETD 2012), Faculty of Engineering and Faculty of Computer Science, Bandar Lampung University, 2012.

2017-12-19

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

Daftar Pustaka

Daftar Pustaka

Daftar Pustaka

Asanilta Fahda, *KenKen Puzzle Solver using Backtracking Algorithm*, Makalah IF2211 Strategi Algoritma - Semester II Tahun 2014/2015, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung 2015.  
Olivia Johanna, Samuel Lukas, Kie Van Ivanky Saputra, *Solving and Modeling Ken-ken Puzzle by Using Hybrid Genetics Algorithm*, 1st International Conference on Engineering and Technology Development (ICETD 2012), Faculty of Engineering and Faculty of Computer Science, Bandar Lampung University, 2012.

# Terima Kasih

- Ada pertanyaan?

2017-12-19

Perbandingan Algoritma *Backtracking* dan Algoritma *Hybrid Genetic* untuk Menyelesaikan Permainan Calcudoku

└ Thank You

└ Terima Kasih

Terima Kasih

■ Ada pertanyaan?