

2016-12-06

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

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

Michael Adrian

2013730039

michaeladrian39@gmail.com

Jurusan Teknik Informatika  
Fakultas Teknologi Informasi dan Sains  
Universitas Katolik Parahyangan

6 Desember 2016

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

Michael Adrian  
2013730039  
michaeladrian39@gmail.com

Jurusan Teknik Informatika  
Fakultas Teknologi Informasi dan Sains  
Universitas Katolik Parahyangan

6 Desember 2016

- Salah satu jenis permainan teka-teki aritmatika dan logika
- Dikenal juga sebagai KenKen, KenDoku, atau Mathdoku
- Diciptakan pada tahun 2004 oleh Tetsuya Miyamoto, seorang guru matematika dari Jepang
- Diciptakan untuk melatih kemampuan matematika dan logika dengan cara yang menyenangkan

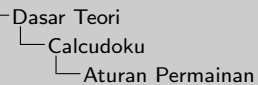
- Salah satu jenis permainan teka-teki aritmatika dan logika
- Dikenal juga sebagai KenKen, KenDoku, atau Mathdoku
- Diciptakan pada tahun 2004 oleh Tetsuya Miyamoto, seorang guru matematika dari Jepang
- Diciptakan untuk melatih kemampuan matematika dan logika dengan cara yang menyenangkan

Sebagai salah satu jenis permainan teka-teki aritmatika dan *grid*, Calcudoku, atau dikenal juga sebagai KenKen, KenDoku, atau Mathdoku, diciptakan pada tahun 2004 oleh seorang guru matematika dari Jepang yang bernama Tetsuya Miyamoto untuk memenuhi tujuannya untuk melatih kemampuan matematika dan logika siswa-siswinya dengan cara yang menyenangkan. Nama KenKen diambil dari kata bahasa Jepang yang berarti kepandaian. Permainan yang mengasah otak ini dengan cepat menyebar ke seluruh Jepang dan Amerika Serikat, menggantikan permainan teka-teki silang di banyak koran. Permainan ini kemudian menjadi sensasi di seluruh dunia setelah munculnya versi *online* dan *mobile* dari permainan teka-teki ini, khususnya menarik untuk pecinta permainan teka-teki angka seperti Sudoku.

- 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*

2016-12-06

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



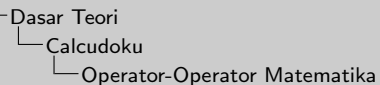
Seperti dalam Sudoku, dalam teka-teki ini, pemain diberikan sebuah *grid* dengan ukuran  $n \times n$ , dengan  $n$  biasanya antara 3 sampai dengan 9. *Grid* ini harus diisi dengan angka 1 sampai dengan  $n$  sehingga dalam setiap baris setiap angka hanya muncul sekali, dalam setiap kolom setiap angka 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  $n \times n$
- $n$  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*

- 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

2016-12-06

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



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)
  - x (perkalian)
  - / (pembagian)
  - = (sama dengan)
- Jika operasi matematika yang ditentukan adalah pengurangan atau pembagian, maka ukuran cage harus berukuran dua sel

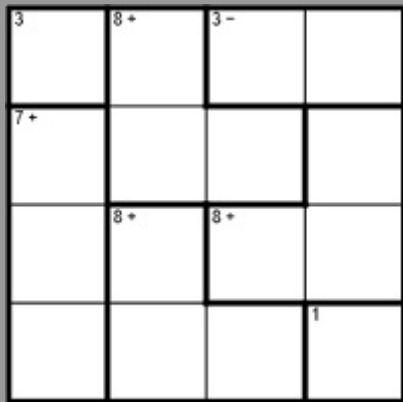


Figure 1: Contoh permainan teka-teki dengan ukuran *grid* 4 x 4 yang belum diselesaikan.



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 Calculudoku yang diberikan pada Gambar 1.

2016-12-06

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

## - Dasar Teori

- Çalcudoku

### —Contoh Solusi

### Contoh Solusi

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

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

- 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*)

2016-12-06

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

## - Dasar Teori

### - 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 satu 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 disebut DFS (Depth First Search)

- Singkatnya, langkah-langkah dasar dari implementasi algoritma *backtracking* dapat dijelaskan sebagai berikut:
  - ① Carilah sel pertama atau sel yang kosong di dalam *grid*
  - ② Isilah sel dengan sebuah angka dimulai dari 1 sampai  $n$  sampai sebuah angka yang berlaku (*valid*) ditemukan atau sampai angka sudah melebihi  $n$
  - ③ Jika angka untuk sel berlaku, ulangi langkah 1 dan 2
  - ④ Jika angka untuk sel sudah melebihi  $n$  dan tidak ada angka dari 1 sampai  $n$  yang berlaku untuk sel tersebut, mundur ke sel sebelumnya dan cobalah kemungkinan angka berikutnya yang berlaku untuk sel tersebut
  - ⑤ Jika tidak ada lagi sel yang kosong, solusi sudah ditemukan

- Singkatnya, langkah-langkah dasar dari implementasi algoritma backtracking dapat dijelaskan sebagai berikut:
  - Carilah set pertama atau set yang kosong di dalam grid
  - Halah sel dengan sebuah angka dimulai dari 1 sampai 9 sampai sebuah angka yang berlaku (valid) ditemukan atau sampai angka sudah melebihi n
  - Jika angka untuk sel berlaku, ulangi langkah 1 dan 2
  - Jika angka untuk sel berlaku melebihi n dan tidak ada angka dari 1 sampai 9 yang berlaku untuk sel tersebut, mundur ke sel sebelumnya dan cobalah kemungkinan angka berikutnya yang berlaku untuk sel tersebut
  - Jika tidak ada lagi sel yang kosong, solusi sudah ditemukan

Singkatnya, langkah-langkah dasar dari implementasi algoritma *backtracking* dapat dijelaskan sebagai berikut [1]:

1. Carilah sel pertama atau sel yang kosong di dalam *grid*.
2. Isilah sel dengan sebuah angka dimulai dari 1 sampai  $n$  sampai sebuah angka yang berlaku (*valid*) ditemukan atau sampai angka sudah melebihi  $n$ .
3. Jika angka untuk sel berlaku, ulangi langkah 1 dan 2.
4. Jika angka untuk sel sudah melebihi  $n$  dan tidak ada angka dari 1 sampai  $n$  yang berlaku untuk sel tersebut, mundur ke sel sebelumnya dan cobalah kemungkinan angka berikutnya yang berlaku untuk sel tersebut.
5. Jika tidak ada lagi sel yang kosong, solusi sudah ditemukan.



- 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

2016-12-06

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

## - Dasar Teori

- 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 *Calculus*
- Jika algoritma sudah tidak bisa menyelesaikan permainan, maka algoritma genetik akan mulai dijalankan

- 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*
  - *Evil twin rule*
  - *Killer combination*
  - *X-wing*

2016-12-06

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

## - Dasar Teori

- 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 Calcudoku
- + Beberapa aturan logika yang digunakan dalam algoritma ini adalah:
  - Single square rule
  - Naked subset rule
  - Hidden single rule
  - Evil twin rule
  - Killer combination
  - X-wing

# Aturan *Single Square*

2016-12-06

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

└ Dasar Teori

└ Algoritma *Hybrid Genetic*└ Aturan *Single Square*

- Digunakan jika sebuah cage hanya berisi satu sel
- Nilai dari sel tersebut sama dengan angka tujuan yang telah ditentukan

- Digunakan jika sebuah cage hanya berisi satu sel
- Nilai dari sel tersebut sama dengan angka tujuan yang telah ditentukan

Aturan *single square* digunakan jika sebuah cage hanya berisi satu sel. Hal ini berarti nilai dari sel tersebut sama dengan angka tujuan yang telah ditentukan.

# Aturan *Naked Subset*

- Digunakan jika ada  $n$  sel dalam kolom atau baris yang sama yang mempunyai  $n$  kemungkinan nilai yang sama persis untuk mengisikannya, dengan  $n \geq 2$ .
- Sel-sel lainnya dalam baris dan kolom tersebut tidak mungkin diisi dengan nilai yang sama dengan nilai milik  $n$  sel tersebut.

2016-12-06

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

### Dasar Teori

#### Algoritma *Hybrid Genetic*

#### Aturan *Naked Subset*

#### Aturan *Naked Subset*

- Digunakan jika ada  $n$  sel dalam kolom atau baris yang sama yang mempunyai  $n$  kemungkinan nilai yang sama persis untuk mengisikannya, dengan  $n \geq 2$ .
- Sel-sel lainnya dalam baris dan kolom tersebut tidak mungkin diisi dengan nilai yang sama dengan nilai milik  $n$  sel tersebut.

Aturan *naked subset* digunakan jika ada  $n$  sel dalam kolom atau baris yang sama yang mempunyai  $n$  kemungkinan nilai yang sama persis untuk mengisikannya, dengan  $n \geq 2$ . Hal ini berarti sel-sel lainnya dalam baris dan kolom tersebut tidak mungkin diisi dengan nilai yang sama dengan nilai milik  $n$  sel tersebut.

3	9	4	17	6	17	18	27	5
---	---	---	----	---	----	----	----	---

Figure 3: Contoh bagaimana cara mendeteksi aturan *naked pair*

- Gambar 3 menunjukkan bagaimana cara kerja aturan ini
- Sel-sel pada kolom ke-4 dan ke-6 mempunyai tepat dua kemungkinan nilai (1 atau 7)
- Ini disebut sebagai *naked pair*
- Karena angka 1 dan 7 harus diisi pada sel-sel pada kolom ke-4 dan ke-6, maka angka 1 dan 7 bisa dieliminasi dari sel-sel pada kolom ke-7 dan ke-8

2016-12-06

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

## - Dasar Teori

- Algoritma *Hybrid Genetic*

- Aturan *Naked Subset*

### Aturan Naked Subset

3	9	4	17	6	17	18	27
---	---	---	----	---	----	----	----

Figure 3: Contoh bagaimana cara mendeteksi aturan *naked pair*

- Gambar 3 menunjukkan bagaimana cara kerja aturan ini
- Sel-sel pada kolom ke-4 dan ke-6 mempunyai tepat dua kemungkinan nilai (1 atau 7)
- Ini disebut sebagai *makud pair*
- Karena angka 1 dan 7 harus diisi pada sel-sel pada kolom ke-4 dan ke-6, maka angka 1 dan 7 bisa dieliminasi dari sel-sel pada kolom ke-7 dan ke-8

Gambar 3 menunjukkan bagaimana cara kerja aturan ini. Sel-sel pada kolom ke-4 dan ke-6 mempunyai tepat dua kemungkinan nilai (1 atau 7). Ini disebut sebagai *naked pair*. Karena angka 1 dan 7 harus diisi pada sel-sel pada kolom ke-4 dan ke-6, maka angka 1 dan 7 bisa dieliminasi dari sel-sel pada kolom ke-7 dan ke-8.

# Aturan *Evil Twin*

- Digunakan jika sebuah *cage* berisikan dua sel, dan salah satu dari kedua sel sudah terisi, maka sel yang satunya lagi diisi dengan angka yang jika kedua angka dihitung dengan operasi matematika yang ditentukan maka akan menghasilkan angka tujuan yang ditentukan
- Bisa digeneralisasikan untuk *cage* yang berukuran lebih dari 2 sel
- Sel yang belum terisi yang terakhir dalam sebuah area diisi oleh sebuah nilai yang diperlukan untuk mencapai nilai tujuan menggunakan operasi matematika yang telah ditentukan.

2016-12-06

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

### Dasar Teori

#### Algoritma *Hybrid Genetic*

#### Aturan *Evil Twin*

#### Aturan *Evil Twin*

- Digunakan jika sebuah *cage* berisikan dua sel, dan salah satu dari kedua sel sudah terisi, maka sel yang satunya lagi diisi dengan angka yang jika kedua angka dihitung dengan operasi matematika yang ditentukan maka akan menghasilkan angka tujuan yang ditentukan
- Bisa digeneralisasikan untuk *cage* yang berukuran lebih dari 2 sel
- Sel yang belum terisi yang terakhir dalam sebuah area diisi oleh sebuah nilai yang diperlukan untuk mencapai nilai tujuan menggunakan operasi matematika yang telah ditentukan.

Aturan *evil twin* digunakan jika sebuah *cage* berisikan dua sel, dan salah satu dari kedua sel sudah terisi, maka sel yang satunya lagi diisi dengan angka yang jika kedua angka dihitung dengan operasi matematika yang ditentukan maka akan menghasilkan angka tujuan yang ditentukan. Aturan ini adalah aturan yang paling mudah. Kenyataannya, aturan ini bisa digeneralisasikan untuk *cage* yang berukuran lebih dari 2 sel. Sel yang belum terisi yang terakhir dalam sebuah area diisi oleh sebuah nilai yang diperlukan untuk mencapai nilai tujuan menggunakan operasi matematika yang telah ditentukan.





Figure 5: Contoh aturan *hidden single*

- Digunakan jika sebuah angka hanya bisa diisikan dalam satu sel dalam sebuah baris atau kolom.
- Pada Gambar 5, nilai-nilai yang mungkin untuk sel yang paling kiri adalah 3, 5, dan 7
- Tetapi dalam baris ini, angka 7 harus muncul dalam salah satu selnya, dan hanya sel yang paling kiri tersebut yang memiliki kemungkinan nilai 7
- Ini disebut sebagai *hidden single*
- Sel tersebut harus diisi dengan angka 7.

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

## - Dasar Teori

- Algoritma *Hybrid Genetic*

- Aturan *Hidden Single*



3	6	4	2	8	9
---	---	---	---	---	---

Figure 5: Contoh aturan hidden singi

- Digunakan jika sebuah angka hanya bisa diisikan dalam satu sel dalam sebuah baris atau kolom.
- Pada Gambar 5, nilai-nilai yang mungkin untuk sel yang paling kiri adalah 3, 5, dan 7
- Tetapi dalam baris ini, angka 7 harus muncul dalam salah satu selnya, dan hanya sel yang paling kiri tersebut yang memiliki kemungkinan nilai 7
- Ini disebut sebagai *hidden single*
- Sel tersebut harus diisi dengan angka 7.

Aturan *hidden single* digunakan jika sebuah angka hanya bisa diisikan dalam satu sel dalam sebuah baris atau kolom. Aturan ini secara konsep cukup mudah, tetapi kadang-kadang sulit untuk diamati. Pada Gambar 5, nilai-nilai yang mungkin untuk sel yang paling kiri adalah 3, 5, dan 7, tetapi dalam baris ini, angka 7 harus muncul dalam salah satu selnya, dan hanya sel yang paling kiri tersebut yang memiliki kemungkinan nilai 7. Ini disebut sebagai *hidden single*. Sel tersebut harus diisi dengan angka 7.



# Aturan *Killer Combination*

- Digunakan jika sebuah *cage* berisikan sel-sel yang berada dalam baris atau kolom yang sama dan operasi yang ditentukan adalah penjumlahan
- Kemungkinan angka yang unik untuk aturan *killer combination* berhubungan dengan ukuran *cage*
- Contoh, jika sebuah *cage* memiliki dua sel dan angka tujuannya adalah 3, maka kemungkinan angka yang bisa diisikan ke dalam kedua sel tersebut adalah 1 atau 2
- Hal ini berarti semua angka lainnya tidak mungkin diisikan ke dalam kedua sel tersebut
- Jika sebuah *cage* memiliki tiga sel dan angka tujuannya adalah 24, maka kemungkinan angka yang bisa diisikan ke dalam ketiga sel tersebut adalah 7, 8, atau 9

2016-12-06

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

### Dasar Teori

#### Algoritma *Hybrid Genetic*

#### Aturan *Killer Combination*

#### Aturan *Killer Combination*

- Digunakan jika sebuah *cage* berisikan sel-sel yang berada dalam baris atau kolom yang sama dan operasi yang ditentukan adalah penjumlahan
- Kemungkinan angka yang unik untuk aturan *killer combination* berhubungan dengan ukuran *cage*
- Contoh, jika sebuah *cage* memiliki dua sel dan angka tujuannya adalah 3, maka kemungkinan angka yang bisa diisikan ke dalam kedua sel tersebut adalah 1 atau 2
- Hal ini berarti semua angka lainnya tidak mungkin diisikan ke dalam kedua sel tersebut
- Jika sebuah *cage* memiliki tiga sel dan angka tujuannya adalah 24, maka kemungkinan angka yang bisa diisikan ke dalam ketiga sel tersebut adalah 7, 8, atau 9

Aturan *killer combination* adalah aturan yang paling krusial. Aturan ini digunakan jika sebuah *cage* berisikan sel-sel yang berada dalam baris atau kolom yang sama dan operasi yang ditentukan adalah penjumlahan. Kemungkinan angka yang unik untuk aturan *killer combination* berhubungan dengan ukuran *cage*. Contoh, jika sebuah *cage* memiliki dua sel dan angka tujuannya adalah 3, maka kemungkinan angka yang bisa diisikan ke dalam kedua sel tersebut adalah 1 atau 2. Hal ini berarti semua angka lainnya tidak mungkin diisikan ke dalam kedua sel tersebut. Contoh lain, jika sebuah *cage* memiliki tiga sel dan angka tujuannya adalah 24, maka kemungkinan angka yang bisa diisikan ke dalam ketiga sel tersebut adalah 7, 8, atau 9.

Cage size	Cage value	Combination
2	3	1/2
2	4	1/3
2	17	8/9
2	16	7/9

Figure 6: Contoh aturan *killer combination* untuk cage dengan ukuran 2 sel

Cage size	Cage value	Combination
2	3	1/2
2	4	1/3
2	17	8/9
2	16	7/9

Figure 6: Contoh aturan killer combination untuk cage dengan ukuran 2

- Digunakan jika hanya ada dua kemungkinan angka yang bisa diisikan ke dalam dua sel yang berada di dalam dua baris yang berbeda, dan dua kemungkinan angka tersebut juga berada di dalam kolom yang sama maka sel-sel lainnya dalam kolom tersebut tidak mungkin diisi oleh dua kemungkinan angka tersebut

4. Digunakan jika hanya ada dua kemungkinan angka yang bisa disisipkan ke dalam dua sel yang berada di dalam dua baris yang berbeda, dan dua kemungkinan angka tersebut juga berada di dalam kolom yang sama maka sel-sel lainnya dalam kolom tersebut tidak mungkin diisi oleh dua kemungkinan angka tersebut

Aturan *X-wing* digunakan jika hanya ada dua kemungkinan angka yang bisa diisikan ke dalam dua sel yang berada di dalam dua baris yang berbeda, dan dua kemungkinan angka tersebut juga berada di dalam kolom yang sama maka sel-sel lainnya dalam kolom tersebut tidak mungkin diisi oleh dua kemungkinan angka tersebut.

# Aturan X-Wing



Figure 7: Contoh aturan X-wing [2]

2016-12-06

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

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Aturan X-Wing

Aturan X-Wing



Figure 7: Contoh aturan X-wing [2]

# Aturan *X-Wing*

- Gambar 7 menampilkan contoh penggunaan aturan *X-wing*
- Misalnya, jika sel A diisi oleh angka 7, maka angka 7 akan dieliminasi dari sel B dan sel C
- Karena sel A dengan sel C dan sel D 'terkunci', maka sel D harus diisi oleh angka 7
- Jadi, angka 7 harus di isi pada sel A dan sel D atau pada sel B dan sel C
- Angka 7 bisa dieliminasi dari sel-sel yang berwarna hijau

2016-12-06

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

### Dasar Teori

#### Algoritma *Hybrid Genetic*

#### Aturan *X-Wing*

#### Aturan *X-Wing*

- Gambar 7 menampilkan contoh penggunaan aturan *X-wing*
- Misalnya, jika sel A diisi oleh angka 7, maka angka 7 akan dieliminasi dari sel B dan sel C
- Karena sel A dengan sel C dan sel D 'terkunci', maka sel D harus diisi oleh angka 7
- Jadi, angka 7 harus di isi pada sel A dan sel D atau pada sel B dan sel C
- Angka 7 bisa dieliminasi dari sel-sel yang berwarna hijau

Gambar 7 menampilkan contoh penggunaan aturan *X-wing*. Misalnya, jika sel A diisi oleh angka 7, maka angka 7 akan dieliminasi dari sel B dan sel C. Karena sel A dengan sel C dan sel D 'terkunci', maka sel D harus diisi oleh angka 7. Jadi, angka 7 harus di isi pada sel A dan sel D atau pada sel B dan sel C. Angka 7 bisa dieliminasi dari sel-sel yang berwarna hijau.

- 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.

2016-12-06

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

## - Dasar Teori

- Algoritma *Hybrid Genetic*

- Algoritma Genetik

## 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.

Algoritma genetik 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

- Meng-*encode* kemungkinan jawaban untuk sebuah masalah yang diberikan.
- Dievaluasi dan diberi *fitness value* berdasarkan seberapa baikkah kromosom dalam menyelesaikan masalah yang diberikan berdasarkan kriteria yang ditentukan
- Nilai kelayakan ini digunakan sebagai probabilitas keberuntungan hidup kromosom dalam satu siklus reproduksi
- *Child chromosome* diproduksi dengan menggabungkan dua *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.

2016-12-06

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

### Dasar Teori

#### Algoritma *Hybrid Genetic*

#### Kromosom

Kromosom ini meng-*encode* kemungkinan jawaban untuk sebuah masalah yang diberikan. Kromosom dievaluasi dan diberi *fitness value* berdasarkan seberapa baikkah kromosom dalam menyelesaikan masalah yang diberikan berdasarkan kriteria yang ditentukan oleh pembuat program. Nilai kelayakan ini digunakan sebagai probabilitas keberuntungan hidup kromosom dalam satu siklus reproduksi. Kromosom baru (kromosom anak, *child chromosome*) diproduksi dengan menggabungkan dua 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.

#### Kromosom

- Meng-*encode* kemungkinan jawaban untuk sebuah masalah yang diberikan.
- Dievaluasi dan diberi *fitness value* berdasarkan seberapa baikkah kromosom dalam menyelesaikan masalah yang diberikan berdasarkan kriteria yang ditentukan
- Nilai kelayakan ini digunakan sebagai probabilitas keberuntungan hidup kromosom dalam satu siklus reproduksi
- *Child chromosome* diproduksi dengan menggabungkan dua *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.

# Cara Kerja Algoritma Genetik

- Cara kerja algoritma genetik adalah sebagai berikut:
  - ① Menentukan populasi kromosom kemungkinan jawaban awal
  - ② Membangkitkan populasi kemungkinan jawaban awal secara acak
  - ③ Mengevaluasi fungsi objektif
  - ④ Melakukan operasi terhadap kromosom menggunakan operator genetik (reproduksi, kawin silang, dan mutasi)
  - ⑤ Ulangi langkah 3 dan 4 sampai mencapai kriteria untuk menghentikan algoritma.

2016-12-06

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

### Dasar Teori

#### Algoritma *Hybrid Genetic*

#### Cara Kerja Algoritma Genetik

- Cara kerja algoritma genetik adalah sebagai berikut:
  - ① Menentukan populasi kromosom kemungkinan jawaban awal
  - ② Membangkitkan populasi kemungkinan jawaban awal secara acak
  - ③ Mengevaluasi fungsi objektif
  - ④ Melakukan operasi terhadap kromosom menggunakan operator genetik (reproduksi, kawin silang, dan mutasi)
  - ⑤ Ulangi langkah 3 dan 4 sampai mencapai kriteria untuk menghentikan algoritma.

Cara kerja algoritma genetik adalah sebagai berikut [2]:

1. Menentukan populasi kromosom kemungkinan jawaban awal.
2. Membangkitkan populasi kemungkinan jawaban awal secara acak.
3. Mengevaluasi fungsi objektif.
4. Melakukan operasi terhadap kromosom menggunakan operator genetik (reproduksi, kawin silang, dan mutasi).
5. Ulangi langkah 3 dan 4 sampai mencapai kriteria untuk menghentikan algoritma.



- Langkah-langkah utama dalam penggunaan algoritma genetik adalah:
  - Membangkitkan populasi kemungkinan jawaban
  - Mencari fungsi objektif dan fungsi kelayakan
  - Penggunaan operator genetik

2016-12-06

## - Dasar Teori

- Algoritma *Hybrid Genetic*

## Langkah-Langkah Utama dalam Penggunaan Algoritma Genetik

Langkah-langkah utama dalam penggunaan algoritma genetik adalah membangkitkan populasi kemungkinan jawaban, mencari fungsi objektif dan fungsi kelayakan, dan penggunaan operator genetik.

- Langkah-langkah utama dalam penggunaan algoritma genetik adalah:
  - Membangkitkan populasi kemungkinan jawaban
  - Mencari fungsi objektif dan fungsi kelayakan
  - Penggunaan operator genetik

# Algoritma Hybrid Genetic

- Pencarian *rule based* dimulai dengan mengasumsikan semua nilai sel yang tidak diketahui dengan semua kemungkinan nilai untuk mengisi sel tersebut tanpa melanggar batasan, dengan  $P(C_{b,k}) = 1, 2, \dots, n$
- Setelah nilai dari satu sel sudah ditentukan, kemungkinan nilai untuk beberapa sel tertentu diperbaharui

2016-12-06

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

### Dasar Teori

#### Algoritma Hybrid Genetic

#### Algoritma Hybrid Genetic

- Pencarian *rule based* dimulai dengan mengasumsikan semua nilai sel yang tidak diketahui dengan semua kemungkinan nilai untuk mengisi sel tersebut tanpa melanggar batasan, dengan  $P(C_{b,k}) = 1, 2, \dots, n$
- Setelah nilai dari satu sel sudah ditentukan, kemungkinan nilai untuk beberapa sel tertentu diperbaharui

Pencarian *rule based* dimulai dengan mengasumsikan semua nilai sel yang tidak diketahui dengan semua kemungkinan nilai untuk mengisi sel tersebut tanpa melanggar batasan, dengan  $P(C_{b,k}) = 1, 2, \dots, n$ . Setelah nilai dari satu sel sudah ditentukan, kemungkinan nilai untuk beberapa sel tertentu diperbaharui.

# Algoritma *Hybrid Genetic*

- Algoritma genetik digunakan saat teka-teki masih tidak bisa diselesaikan setelah mengerjakan semua aturan logika secara berulang-ulang
- Algoritma ini dimulai dengan meng-*encode* kromosom

2016-12-06

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

### Dasar Teori

#### Algoritma *Hybrid Genetic*

#### Algoritma *Hybrid Genetic*

- Algoritma genetik digunakan saat teka-teki masih tidak bisa diselesaikan setelah mengerjakan semua aturan logika secara berulang-ulang
- Algoritma ini dimulai dengan meng-*encode* kromosom

Algoritma genetik digunakan saat teka-teki masih tidak bisa diselesaikan setelah mengerjakan semua aturan logika secara berulang-ulang. Algoritma ini dimulai dengan meng-*encode* kromosom.

# Pemodelan Kromosom

- Satu kromosom terdiri dari  $k$  segmen, dengan  $m \leq n$
- Satu segmen berisikan sekumpulan gen yang belum diselesaikan yang berada di dalam segmen tersebut
- Sebuah segmen merepresentasikan sebuah baris
- Dalam sebuah kromosom, segmen diurutkan dari baris yang paling atas ke baris yang paling bawah
- Setiap segmen merepresentasikan sebuah baris yang belum terselesaikan

2016-12-06

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

### Dasar Teori

#### Algoritma *Hybrid Genetic*

#### Pemodelan Kromosom

- Satu kromosom terdiri dari  $k$  segmen, dengan  $m \leq n$
- Satu segmen berisikan sekumpulan gen yang belum diselesaikan yang berada di dalam segmen tersebut
- Sebuah segmen merepresentasikan sebuah baris
- Dalam sebuah kromosom, segmen diurutkan dari baris yang paling atas ke baris yang paling bawah
- Setiap segmen merepresentasikan sebuah baris yang belum terselesaikan

Satu kromosom terdiri dari  $k$  segmen, dengan  $m \leq n$ . Satu segmen berisikan sekumpulan gen yang belum diselesaikan yang berada di dalam segmen tersebut.

Sebuah segmen merepresentasikan sebuah baris. Dalam sebuah kromosom, segmen diurutkan dari baris yang paling atas ke baris yang paling bawah. Setiap segmen merepresentasikan sebuah baris yang belum terselesaikan.

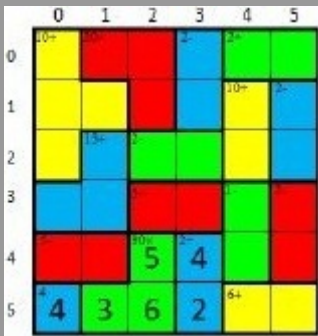


Figure 8: Contoh permainan teka-teki Calcudoku dengan ukuran *grid* 6 x 6

Contoh salah satu kromosom dari permainan teka-teki Calcudoku pada Gambar 8 adalah:

34 35 | 28 29 24 25 | 0 4 5 1 2 3 | 11 6 9 7 8 10 | ◀ ▶ 🔍 ↺ ↻

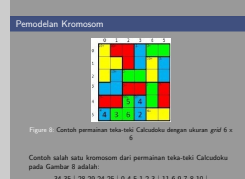
2016-12-06

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

## - Dasar Teori

- Algoritma *Hybrid Genetic*

## – Pemodelan Kromosom



Contoh, salah satu kromosom dari permainan teka-teki Calcudoku pada Gambar 8 adalah

34 35 | 28 29 24 25 | 0 4 5 1 2 3 | 11 6 9 7 8 10 | 12 14 15 17 16 13 | 20 18 19 23 21 22.

# Kawin Silang

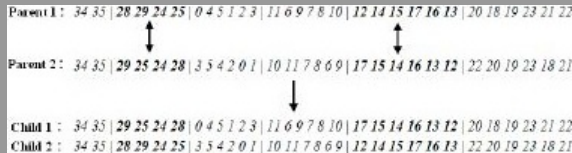


Figure 9: Contoh proses kawin silang antara dua kromosom

- Dua kromosom, yaitu kromosom orang tua, disilangkan untuk membuat dua kromosom yang baru, yaitu kromosom anak, dengan metodologi kawin silang *N-segments*
- Gambar 9 menggambarkan contoh proses kawin silang antara dua kromosom.

2016-12-06

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

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Kawin Silang

### Kawin Silang



Figure 9: Contoh proses kawin silang antara dua kromosom

- Dua kromosom, yaitu kromosom orang tua, disilangkan untuk membuat dua kromosom yang baru, yaitu kromosom anak, dengan metodologi kawin silang *N-segments*
- Gambar 9 menggambarkan contoh proses kawin silang antara dua kromosom.

Dalam proses reproduksi kawin silang, dua kromosom, yaitu kromosom orang tua, disilangkan untuk membuat dua kromosom yang baru, yaitu kromosom anak, dengan metodologi kawin silang *N-segments*. Gambar 9 menggambarkan contoh proses kawin silang antara dua kromosom.

# Mutasi

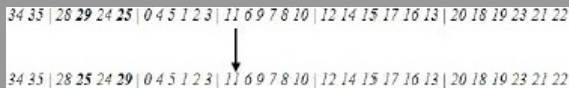


Figure 10: Contoh proses mutasi

- Digunakan untuk mendapatkan kemungkinan kromosom yang lain
- Dilakukan di antara gen yang berada dalam segmen yang sama
- Gambar 10 adalah contoh proses mutasi antara dua gen dalam segmen yang sama

2016-12-06

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

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Mutasi

Mutasi



Figure 10: Contoh proses mutasi

- Digunakan untuk mendapatkan kemungkinan kromosom yang lain
- Dilakukan di antara gen yang berada dalam segmen yang sama
- Gambar 10 adalah contoh proses mutasi antara dua gen dalam segmen yang sama

Pertukaran mutasi digunakan untuk mendapatkan kemungkinan kromosom yang lain.

Mutasi dilakukan di antara gen yang berada dalam segmen yang sama. Gambar 10

adalah contoh proses mutasi antara dua gen dalam segmen yang sama.

# Cara Kerja Algoritma Hybrid Genetic

- Cara kerja algoritma *hybrid genetic* adalah sebagai berikut:
  - Masukkan teka-teki yang akan diselesaikan sebagai input.
  - Program akan merepresentasikan input yang dimasukkan dalam format teka-teki.
  - Program akan mencoba menyelesaikan teka-teki tersebut dengan menggunakan algoritma *rule based* terlebih dahulu.
  - Jika program berhasil menyelesaikan teka-teki tersebut dengan menggunakan algoritma *rule based*, maka algoritma selesai.
  - Jika program gagal dengan menggunakan algoritma *rule based*, maka program akan mencoba menyelesaikan teka-teki tersebut dengan menggunakan algoritma genetik.
  - Jika program berhasil menyelesaikan teka-teki tersebut dengan menggunakan algoritma genetik, maka algoritma selesai.
  - Jika program gagal dalam menyelesaikan teka-teki tersebut setelah menggunakan algoritma genetik, artinya algoritma gagal dalam menyelesaikan teka-teki tersebut.

2016-12-06

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

### Dasar Teori

#### Algoritma Hybrid Genetic

#### Cara Kerja Algoritma Hybrid Genetic

#### Cara Kerja Algoritma Hybrid Genetic

- Cara kerja algoritma *hybrid genetic* adalah sebagai berikut:
  - Masukkan teka-teki yang akan diselesaikan sebagai input.
  - Program akan merepresentasikan input yang dimasukkan dalam format teka-teki.
  - Program akan mencoba menyelesaikan teka-teki tersebut dengan menggunakan algoritma *rule based* terlebih dahulu.
  - Jika program berhasil menyelesaikan teka-teki tersebut dengan menggunakan algoritma *rule based*, maka algoritma selesai.
  - Jika program gagal dengan menggunakan algoritma *rule based*, maka program akan mencoba menyelesaikan teka-teki tersebut dengan menggunakan algoritma genetik.
  - Jika program berhasil menyelesaikan teka-teki tersebut dengan menggunakan algoritma genetik, maka algoritma selesai.
  - Jika program gagal dalam menyelesaikan teka-teki tersebut setelah menggunakan algoritma genetik, artinya algoritma gagal dalam menyelesaikan teka-teki tersebut.

Cara kerja algoritma *hybrid genetic* menurut Johanna, Lukas, dan Saputra adalah sebagai berikut [2]:

- Masukkan teka-teki yang akan diselesaikan sebagai input.
- Program akan merepresentasikan input yang dimasukkan dalam format teka-teki.
- Program akan mencoba menyelesaikan teka-teki tersebut dengan menggunakan algoritma *rule based* terlebih dahulu.
- Jika program berhasil menyelesaikan teka-teki tersebut dengan menggunakan algoritma *rule based*, maka algoritma selesai.
- Jika program gagal dengan menggunakan algoritma *rule based*, maka program akan mencoba menyelesaikan teka-teki tersebut dengan menggunakan algoritma genetik.
- Jika program berhasil menyelesaikan teka-teki tersebut dengan menggunakan algoritma genetik, maka algoritma selesai.
- Jika program gagal dalam menyelesaikan teka-teki tersebut setelah menggunakan algoritma genetik, artinya algoritma gagal dalam menyelesaikan teka-teki tersebut.



# Alur Cara Kerja Algoritma Hybrid Genetic

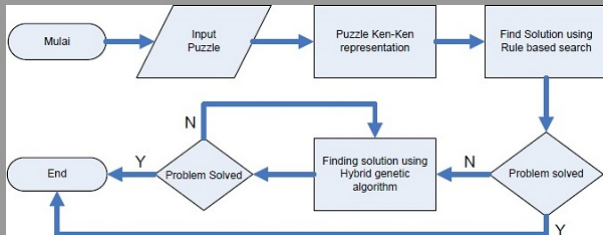


Figure 11: Alur penyelesaian permainan teka-teki Calcudoku dengan menggunakan algoritma *hybrid genetic*

- Alur (*flow chart*) penyelesaian permainan teka-teki Calcudoku dengan menggunakan algoritma *hybrid genetic* dapat dilihat di Gambar 11.

2016-12-06

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

└ Dasar Teori

└ Algoritma *Hybrid Genetic*

└ Alur Cara Kerja Algoritma *Hybrid Genetic*



Figure 11: Alur penyelesaian permainan teka-teki Calcudoku dengan menggunakan algoritma *hybrid genetic*

• Alur (*flow chart*) penyelesaian permainan teka-teki Calcudoku dengan menggunakan algoritma *hybrid genetic* dapat dilihat di Gambar 11.

Alur (*flow chart*) penyelesaian permainan teka-teki Calcudoku dengan menggunakan algoritma *hybrid genetic* dapat dilihat di Gambar 11.

# Analisis Algoritma *Backtracking*

3+	1	8+
3	3+	

Figure 12: Contoh permainan teka-teki Calcudoku dengan ukuran *grid* 3 x 3

- Untuk mengilustrasikan cara kerja algoritma *backtracking*, teka-teki Calcudoku yang digambarkan pada Gambar 12 akan digunakan.
- Contoh teka-teki ini dapat dilihat di Bab 2 (Dasar Teori).

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

2016-12-06

└ Analisis

└ Algoritma *Backtracking*└ Analisis Algoritma *Backtracking*

3+	1	8+
3	3+	

Figure 12: Contoh permainan teka-teki Calcudoku dengan ukuran *grid* 3 x 3

- Untuk mengilustrasikan cara kerja algoritma *backtracking*, teka-teki Calcudoku yang digambarkan pada Gambar 12 akan digunakan.
- Contoh teka-teki ini dapat dilihat di Bab 2 (Dasar Teori).

Untuk mengilustrasikan cara kerja algoritma *backtracking*, teka-teki Calcudoku yang digambarkan pada Gambar 12 akan digunakan.

# Analisis Algoritma *Backtracking*

- Algoritma *backtracking* dimulai dengan teka-teki yang belum diselesaikan (*state* 1).
- Algoritma mengisi sel pada baris ke-1 dan kolom ke-1 dengan angka 1 (*state* 2). Algoritma lalu maju ke sel berikutnya.
- Algoritma lalu mengisi sel pada baris ke-1 dan kolom ke-2 dengan angka 1 (*state* 3), tetapi angka 1 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 4), tetapi angka 2 tidak sesuai dengan angka tujuan dari *cage* tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 3 (*state* 5), tetapi angka 3 tidak sesuai dengan angka tujuan dari *cage* tersebut.
- *State* 3, *state* 4, dan *state* 5 digambarkan pada Gambar 13.

2016-12-06

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

### Analisis

#### Algoritma *Backtracking*

#### Analisis Algoritma *Backtracking*

- Algoritma *backtracking* dimulai dengan teka-teki yang belum diselesaikan (*state* 1).
- Algoritma mengisi sel pada baris ke-1 dan kolom ke-1 dengan angka 1 (*state* 2). Algoritma lalu maju ke sel berikutnya.
- Algoritma lalu mengisi sel pada baris ke-1 dan kolom ke-2 dengan angka 1 (*state* 3), tetapi angka 1 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 4), tetapi angka 2 tidak sesuai dengan angka tujuan dari *cage* tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 3 (*state* 5), tetapi angka 3 tidak sesuai dengan angka tujuan dari *cage* tersebut.
- *State* 3, *state* 4, dan *state* 5 digambarkan pada Gambar 13.

#### Analisis Algoritma *Backtracking*

- Algoritma *backtracking* dimulai dengan teka-teki yang belum diselesaikan (*state* 1).
- Algoritma mengisi sel pada baris ke-1 dan kolom ke-1 dengan angka 1 (*state* 2). Algoritma lalu maju ke sel berikutnya.
- Algoritma lalu mengisi sel pada baris ke-1 dan kolom ke-2 dengan angka 1 (*state* 3), tetapi angka 1 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 4), tetapi angka 2 tidak sesuai dengan angka tujuan dari *cage* tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 3 (*state* 5), tetapi angka 3 tidak sesuai dengan angka tujuan dari *cage* tersebut.
- *State* 3, *state* 4, dan *state* 5 digambarkan pada Gambar 13.

# Analisis Algoritma *Backtracking*



Figure 13: Ilustrasi *state* 3, 4, dan 5 pada sebuah *grid* teka-teki Calcudoku

2016-12-06

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

└ Analisis

└ Algoritma *Backtracking*

└ Analisis Algoritma *Backtracking*



Figure 13: Ilustrasi *state* 3, 4, dan 5 pada sebuah *grid* teka-teki Calcudoku

# Analisis Algoritma *Backtracking*

- Karena semua kemungkinan angka untuk baris ke-1 dan kolom ke-2 telah dicoba dan gagal, maka algoritma harus mundur kembali ke (*state* 1). Algoritma mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 6). Algoritma lalu maju ke sel berikutnya.
- Algoritma mengisi sel pada baris ke-1 dan kolom ke-2 dengan angka 1 (*state* 7), dan ternyata hasilnya sesuai dengan angka tujuan dari *cage* tersebut. Algoritma lalu maju ke sel berikutnya.
- Algoritma mengisi sel pada baris ke-1 dan kolom ke-3 dengan angka 1 (*state* 8), tetapi angka 1 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 9), tetapi angka 2 sudah pernah digunakan dalam baris tersebut.

2016-12-06

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

### Analisis

#### Algoritma *Backtracking*

#### Analisis Algoritma *Backtracking*

- Karena semua kemungkinan angka untuk baris ke-1 dan kolom ke-2 telah dicoba dan gagal, maka algoritma harus mundur kembali ke (*state* 1). Algoritma mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 6). Algoritma lalu maju ke sel berikutnya.
- Algoritma mengisi sel pada baris ke-1 dan kolom ke-2 dengan angka 1 (*state* 7), dan ternyata hasilnya sesuai dengan angka tujuan dari *cage* tersebut. Algoritma lalu maju ke sel berikutnya.
- Algoritma mengisi sel pada baris ke-1 dan kolom ke-3 dengan angka 1 (*state* 8), tetapi angka 1 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 9), tetapi angka 2 sudah pernah digunakan dalam baris tersebut.

- Karena semua kemungkinan angka untuk baris ke-1 dan kolom ke-2 telah dicoba dan gagal, maka algoritma harus mundur kembali ke (*state* 1). Algoritma mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 6). Algoritma lalu maju ke sel berikutnya.
- Algoritma mengisi sel pada baris ke-1 dan kolom ke-2 dengan angka 1 (*state* 7), dan ternyata hasilnya sesuai dengan angka tujuan dari *cage* tersebut. Algoritma lalu maju ke sel berikutnya.
- Algoritma mengisi sel pada baris ke-1 dan kolom ke-3 dengan angka 1 (*state* 8), tetapi angka 1 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 9), tetapi angka 2 sudah pernah digunakan dalam baris tersebut.

# Analisis Algoritma *Backtracking*

- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 3 (*state* 10). Algoritma telah selesai mengisi baris ke-1, sehingga bisa maju ke baris berikutnya.
- Algoritma mengisi sel pada baris ke-2 dan kolom ke-1 dengan angka 1 (*state* 11), dan ternyata hasilnya sesuai dengan angka tujuan dari *cage* tersebut. Algoritma lalu maju ke sel berikutnya.
- Algoritma mengisi sel pada baris ke-2 dan kolom ke-2 dengan angka 1 (*state* 12), tetapi angka 1 sudah pernah digunakan dalam baris dan kolom tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 13). Algoritma lalu maju ke sel berikutnya.

2016-12-06

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

### Analisis

#### Algoritma *Backtracking*

#### Analisis Algoritma *Backtracking*

- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 3 (*state* 10). Algoritma telah selesai mengisi baris ke-1, sehingga bisa maju ke baris berikutnya.
- Algoritma mengisi sel pada baris ke-2 dan kolom ke-1 dengan angka 1 (*state* 11), dan ternyata hasilnya sesuai dengan angka tujuan dari *cage* tersebut. Algoritma lalu maju ke sel berikutnya.
- Algoritma mengisi sel pada baris ke-2 dan kolom ke-2 dengan angka 1 (*state* 12), tetapi angka 1 sudah pernah digunakan dalam baris dan kolom tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 13). Algoritma lalu maju ke sel berikutnya.

- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 3 (*state* 10). Algoritma telah selesai mengisi baris ke-1, sehingga bisa maju ke baris berikutnya.
- Algoritma mengisi sel pada baris ke-2 dan kolom ke-1 dengan angka 1 (*state* 11), dan ternyata hasilnya sesuai dengan angka tujuan dari *cage* tersebut. Algoritma lalu maju ke sel berikutnya.
- Algoritma mengisi sel pada baris ke-2 dan kolom ke-2 dengan angka 1 (*state* 12), tetapi angka 1 sudah pernah digunakan dalam baris dan kolom tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 13). Algoritma lalu maju ke sel berikutnya.

# Analisis Algoritma *Backtracking*

- Algoritma mengisi sel pada baris ke-2 dan kolom ke-3 dengan angka 1 (*state* 14), tetapi angka 1 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 15), tetapi angka 2 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 3 (*state* 16), tetapi angka 3 sudah pernah digunakan dalam kolom tersebut.
- Karena semua kemungkinan angka untuk baris ke-2 dan kolom ke-3 telah dicoba dan gagal, maka algoritma harus mundur kembali ke (*state* 13). Algoritma mencoba kemungkinan angka berikutnya, yaitu angka 3 (*state* 17). Algoritma lalu maju ke sel berikutnya.

2016-12-06

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

### Analisis

#### Algoritma *Backtracking*

#### Analisis Algoritma *Backtracking*

- Algoritma mengisi sel pada baris ke-2 dan kolom ke-3 dengan angka 1 (*state* 14), tetapi angka 1 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 15), tetapi angka 2 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 3 (*state* 16), tetapi angka 3 sudah pernah digunakan dalam kolom tersebut.
- Karena semua kemungkinan angka untuk baris ke-2 dan kolom ke-3 telah dicoba dan gagal, maka algoritma harus mundur kembali ke (*state* 13). Algoritma mencoba kemungkinan angka berikutnya, yaitu angka 3 (*state* 17). Algoritma lalu maju ke sel berikutnya.

#### Analisis Algoritma *Backtracking*

- Algoritma mengisi sel pada baris ke-2 dan kolom ke-3 dengan angka 1 (*state* 14), tetapi angka 1 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 15), tetapi angka 2 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 3 (*state* 16), tetapi angka 3 sudah pernah digunakan dalam kolom tersebut.
- Karena semua kemungkinan angka untuk baris ke-2 dan kolom ke-3 telah dicoba dan gagal, maka algoritma harus mundur kembali ke (*state* 13). Algoritma mencoba kemungkinan angka berikutnya, yaitu angka 3 (*state* 17). Algoritma lalu maju ke sel berikutnya.

# Analisis Algoritma *Backtracking*

- Algoritma mengisi sel pada baris ke-2 dan kolom ke-3 dengan angka 1 (*state* 18), tetapi angka 1 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 19), dan ternyata hasilnya sesuai dengan angka tujuan dari *cage* tersebut, seperti digambarkan pada Gambar 14. Algoritma telah selesai mengisi baris ke-2, sehingga bisa maju ke baris berikutnya.

2016-12-06

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

### Analisis

#### Algoritma *Backtracking*

#### Analisis Algoritma *Backtracking*

- Algoritma mengisi sel pada baris ke-2 dan kolom ke-3 dengan angka 1 (*state* 18), tetapi angka 1 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 19), dan ternyata hasilnya sesuai dengan angka tujuan dari *cage* tersebut, seperti digambarkan pada Gambar 14. Algoritma telah selesai mengisi baris ke-2, sehingga bisa maju ke baris berikutnya.

- Algoritma mengisi sel pada baris ke-2 dan kolom ke-3 dengan angka 1 (*state* 18), tetapi angka 1 sudah pernah digunakan dalam baris tersebut.
- Algoritma lalu mencoba kemungkinan angka berikutnya, yaitu angka 2 (*state* 19), dan ternyata hasilnya sesuai dengan angka tujuan dari *cage* tersebut, seperti digambarkan pada Gambar 14. Algoritma telah selesai mengisi baris ke-2, sehingga bisa maju ke baris berikutnya.



3+ 2	1 1	8+ 3
1	3	2
3	3+	

Figure 14: Ilustrasi *state* 19 pada sebuah *grid* teka-teki Calcudoku

2016-12-06

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

## - Analisis

- Algoritma *Backtracking*

### – Analisis Algoritma *Backtracking*

3+	1	3+
2	1	3
1	3	2
3	3+	

Figure 14: Ilustrasi state 19 pada sebuah grid teka-teki Calcudoku

# Analisis Algoritma *Backtracking*

- Algoritma terus bekerja hingga akhirnya menemukan solusinya pada *state* 25. *State space tree* ini telah mencapai simpul tujuannya, yaitu simpul 25, dengan jalur 2-1-3-1-3-2-3-2-1, seperti digambarkan pada Gambar 16.

2016-12-06

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

### Analisis

#### Algoritma *Backtracking*

#### Analisis Algoritma *Backtracking*

- Algoritma terus bekerja hingga akhirnya menemukan solusinya pada *state* 25. *State space tree* ini telah mencapai simpul tujuannya, yaitu simpul 25, dengan jalur 2-1-3-1-3-2-3-2-1, seperti digambarkan pada Gambar 16.

- Algoritma terus bekerja hingga akhirnya menemukan solusinya pada *state* 25.
- Gambar 15 menggambarkan *state* 25 dalam penyelesaian teka-teki Calcudoku ini. Algoritma ini mencapai solusinya pada *state* 25, seperti pada *state space tree* yang digambarkan dalam Gambar 16. *State space tree* ini telah mencapai simpul tujuannya, yaitu simpul 25, dengan jalur 2-1-3-1-3-2-3-2-1.

<sup>3+</sup> 2	<sup>1</sup> 1	<sup>8+</sup> 3
1	3	2
<sup>3</sup> 3	<sup>3+</sup> 2	1

Figure 15: *State* 25, simpul tujuan, sebagai hasil yang dicapai

2016-12-06

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

## - Analisis

### - Algoritma *Backtracking*

### – Analisis Algoritma *Backtracking*

<sup>3+</sup> 2	<sup>1</sup> 1	<sup>8+</sup> 3
1	3	2
<sup>3</sup> 3	<sup>3+</sup> 2	1

Figure 15: State 25, simpul tujuan, sebagai hasil yang dicapai

# Analisis Algoritma *Backtracking*

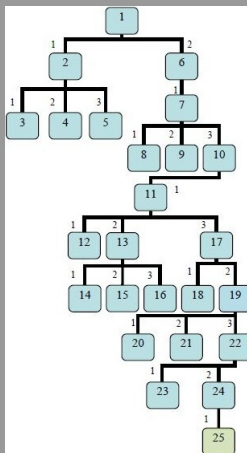


Figure 16: *State space tree* yang dikembangkan dalam proses menyelesaikan teka-teki Calcudoku yang digambarkan pada Gambar 12

2016-12-06

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

Analisis

Algoritma *Backtracking*

Analisis Algoritma *Backtracking*


Analisis Algoritma *Backtracking*

Figure 16: *State space tree* yang dikembangkan dalam proses menyelesaikan teka-teki Calcudoku yang digambarkan pada Gambar 12

# Analisis Algoritma Hybrid Genetic

4-	1-		1-	15+	
	30*				1-
2-	2/		1-	3/	
		24*			1
5+			7+	60*	
	1-				

Figure 17: Contoh permainan teka-teki Calcudoku dengan ukuran *grid* 6 x 6 yang belum diselesaikan

- Untuk mengilustrasikan cara kerja algoritma *hybrid genetic*, 

2016-12-06

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

Analisis

Algoritma *Hybrid Genetic*

Analisis Algoritma *Hybrid Genetic*

1	2	3	4	5	6
1	2	3	4	5	6
1	2	3	4	5	6
1	2	3	4	5	6
1	2	3	4	5	6
1	2	3	4	5	6

Figure 17: Contoh permainan teka-teki Calcudoku dengan ukuran *grid* 6 x 6 yang belum diselesaikan

• Untuk mengilustrasikan cara kerja algoritma *hybrid genetic*.

Untuk mengilustrasikan cara kerja algoritma *hybrid genetic*, akan digunakan permainan teka-teki Calcudoku yang digambarkan pada Gambar 17 sebagai contoh.

# Analisis Algoritma *Hybrid Genetic*

- Algoritma *hybrid genetic* dimulai dengan mencoba menyelesaikan permainan teka-teki Calcudoku dengan algoritma *rule based*
- Sel pada baris ke-4 dan kolom ke-6 adalah bagian dari sebuah *cage* yang berukuran hanya 1 sel
- Oleh karena itu, angka tujuan dari sel tersebut adalah angka tujuan dari *cage* tersebut (aturan *single square*)
- Angka tujuan dari *cage* tersebut adalah 1
- Oleh karena itu sel tersebut dapat langsung diisi dengan angka 1, seperti dapat dilihat pada Gambar 18.

2016-12-06

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

### Analisis

#### Algoritma *Hybrid Genetic*

#### Analisis Algoritma *Hybrid Genetic*

- Algoritma *hybrid genetic* dimulai dengan mencoba menyelesaikan permainan teka-teki Calcudoku dengan algoritma *rule based*
- Sel pada baris ke-4 dan kolom ke-6 adalah bagian dari sebuah *cage* yang berukuran hanya 1 sel
- Oleh karena itu, angka tujuan dari sel tersebut adalah angka tujuan dari *cage* tersebut (aturan *single square*)
- Angka tujuan dari *cage* tersebut adalah 1
- Oleh karena itu sel tersebut dapat langsung diisi dengan angka 1, seperti dapat dilihat pada Gambar 18.

Algoritma *hybrid genetic* dimulai dengan mencoba menyelesaikan permainan teka-teki Calcudoku dengan algoritma *rule based*. Sel pada baris ke-4 dan kolom ke-6 adalah bagian dari sebuah *cage* yang berukuran hanya 1 sel, dan oleh karena itu, angka tujuan dari sel tersebut adalah angka tujuan dari *cage* tersebut (aturan *single square*). Angka tujuan dari *cage* tersebut adalah 1, dan oleh karena itu sel tersebut dapat langsung diisi dengan angka 1, seperti dapat dilihat pada Gambar 18.

4-	1-		1-	15+	
	30*				1-
2-	2/		1-	3/	
		24*			<sup>1</sup> 1
5+			7+	60*	
	1-				

Figure 18: Permainan teka-teki Calcudoku setelah diselesaikan dengan algoritma *rule based*

2016-12-06

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

## - Analisis

- Algoritma *Hybrid Genetic*

- Analisis Algoritma *Hybrid Genetic*

6'	9'		3'	18'
	20'			1'
2'	3'		3'	
		20'		1'
18'			2'	60'
	3'			

Figure 18: Permainan teka-teki Calcudoku setelah diselesaikan dengan algoritma *rule based*

# Analisis Algoritma *Hybrid Genetic*

- Sayangnya, algoritma *rule based* gagal dalam mengisi sel-sel lainnya berdasarkan aturan-aturan yang telah didefinisikan setelah beberapa kali percobaan
- Oleh karena itu, algoritma *hybrid genetic* akan mencoba menyelesaikan teka-teki Calcudoku dengan algoritma genetik

2016-12-06

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

### Analisis

#### Algoritma *Hybrid Genetic*

#### Analisis Algoritma *Hybrid Genetic*

- Sayangnya, algoritma *rule based* gagal dalam mengisi sel-sel lainnya berdasarkan aturan-aturan yang telah didefinisikan setelah beberapa kali percobaan
- Oleh karena itu, algoritma *hybrid genetic* akan mencoba menyelesaikan teka-teki Calcudoku dengan algoritma genetik

Sayangnya, algoritma *rule based* gagal dalam mengisi sel-sel lainnya berdasarkan aturan-aturan yang telah didefinisikan setelah beberapa kali percobaan, sehingga algoritma *hybrid genetic* akan mencoba menyelesaikan teka-teki Calcudoku dengan algoritma genetik.



# Analisis Algoritma Hybrid Genetic

- Dalam contoh ini, parameter-parameter untuk algoritma genetik yang akan digunakan untuk teka-teki Calcudoku ini ditunjukkan pada Tabel 1.
- Untuk mengilustrasikan cara kerja algoritma genetik, hanya 2 generasi pertama yang akan dibahas.

2016-12-06

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

└ Analisis

└ Algoritma Hybrid Genetic

└ Analisis Algoritma Hybrid Genetic

- Dalam contoh ini, parameter-parameter untuk algoritma genetik yang akan digunakan untuk teka-teki Calcudoku ini ditunjukkan pada Tabel 1.
- Untuk mengilustrasikan cara kerja algoritma genetik, hanya 2 generasi pertama yang akan dibahas.

Dalam contoh ini, parameter-parameter untuk algoritma genetik yang akan digunakan untuk teka-teki Calcudoku ini ditunjukkan pada Tabel 1. Setiap generasi terdiri dari 12 kromosom.  $40\% \times 12 \approx 5$  kromosom diambil dari generasi sebelumnya (*elitism*).  $50\% \times 12 \approx 6$  kromosom adalah hasil dari pembentukan kromosom-kromosom baru dengan operasi kawin silang, dan  $10\% \times 12 \approx 1$  kromosom adalah hasil dari pembentukan kromosom-kromosom baru dengan operasi mutasi. Biasanya, operasi kawin silang akan menghasilkan 2 kromosom baru, tetapi dalam kasus ini ditentukan bahwa operasi kawin silang hanya akan menghasilkan 1 kromosom baru. Untuk mengilustrasikan cara kerja algoritma genetik, hanya 2 generasi pertama yang akan dibahas.

Parameter	Nilai
Ukuran Populasi	12
Probabilitas <i>Elitism</i>	40%
Probabilitas Kawin Silang	50%
Probabilitas Mutasi	10%

Table 1: Tabel parameter untuk algoritma genetik yang akan digunakan untuk menyelesaikan teka-teki Calcudoku yang digambarkan pada Gambar 18

Parameter	Nilai
Ukuran Populasi	120
Probabilitas Elitisme	40%
Probabilitas Kawan Silang	50%
Probabilitas Mutasi	10%

Table 1: Tabel parameter untuk algoritma genetik yang akan digunakan untuk menyelesaikan teka-teki Caludoku yang digambarkan pada Gambar 18

# Analisis Algoritma *Hybrid Genetic*

- Algoritma genetik dimulai dengan membangkitkan kromosom-kromosom baru sebanyak ukuran populasi yang telah ditentukan.
- Dalam contoh ini, ukuran populasi adalah 12
- Maka, algoritma akan membangkitkan 12 kromosom baru
- Ke-12 kromosom awal ini adalah bagian dari generasi pertama.

2016-12-06

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

### Analisis

#### Algoritma *Hybrid Genetic*

#### Analisis Algoritma *Hybrid Genetic*

- Algoritma genetik dimulai dengan membangkitkan kromosom-kromosom baru sebanyak ukuran populasi yang telah ditentukan.
- Dalam contoh ini, ukuran populasi adalah 12
- Maka, algoritma akan membangkitkan 12 kromosom baru
- Ke-12 kromosom awal ini adalah bagian dari generasi pertama.

Algoritma genetik dimulai dengan membangkitkan kromosom-kromosom baru sebanyak ukuran populasi yang telah ditentukan. Dalam contoh ini, ukuran populasi adalah 12, maka algoritma akan membangkitkan 12 kromosom baru. Ke-12 kromosom awal ini adalah bagian dari generasi pertama.

- Berdasarkan nilai kelayakan untuk kromosom-kromosom pada Generasi ke-1 yang ditampilkan pada Tabel 2, 5 kromosom terbaik akan diambil untuk menjadi bagian dari Generasi ke-2
- Ke-5 kromosom yang terpilih adalah Kromosom 12, Kromosom 5, Kromosom 7, Kromosom 11, dan Kromosom 1

- Berdasarkan nilai kelayakan untuk kromosom-kromosom pada Generasi ke-1 yang ditampilkan pada Tabel 2, 5 kromosom terbaik akan diambil untuk menjadi bagian dari Generasi ke-2. Ke-5 kromosom yang terpilih adalah Kromosom 12, Kromosom 5, Kromosom 7, Kromosom 11, dan Kromosom 1

Berdasarkan nilai kelayakan untuk kromosom-kromosom pada Generasi ke-1 yang ditampilkan pada Tabel 2, 5 kromosom terbaik akan diambil untuk menjadi bagian dari Generasi ke-2. Ke-5 kromosom yang terpilih adalah Kromosom 12, Kromosom 5, Kromosom 7, Kromosom 11, dan Kromosom 1.

# Analisis Algoritma *Hybrid Genetic*

Nomor Kromosom	Nilai Kelayakan
1	0,3333
2	0,3056
3	0,25
4	0,2222
5	0,4444
6	0,1389
7	0,3889
8	0,25
9	0,1389
10	0,3056
11	0,3889
12	0,5556

Table 2: Tabel nilai kelayakan untuk kromosom-kromsوم pada Generasi ke-1

2016-12-06

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

└ Analisis

└ Algoritma *Hybrid Genetic*

└ Analisis Algoritma *Hybrid Genetic*

Nomor Kromosom	Nilai Kelayakan
1	0.3333
2	0.3056
3	0.25
4	0.2222
5	0.4444
6	0.1389
7	0.3889
8	0.25
9	0.1389
10	0.3056
11	0.3889
12	0.5556

Table 2: Tabel nilai kelayakan untuk kromosom-kromsوم pada Generasi ke-1

# Analisis Algoritma Hybrid Genetic

- Untuk Generasi ke-2, 5 kromosom adalah 5 kromosom terbaik dari Generasi ke-1
- 6 kromosom adalah hasil kawin silang dari 2 kromosom dari Generasi ke-1
- 1 kromosom adalah hasil mutasi dari 1 kromosom dari Generasi ke-1

2016-12-06

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

- Analisis

- Algoritma Hybrid Genetic

- Analisis Algoritma Hybrid Genetic

- Untuk Generasi ke-2, 5 kromosom adalah 5 kromosom terbaik dari Generasi ke-1
- 6 kromosom adalah hasil kawin silang dari 2 kromosom dari Generasi ke-1
- 1 kromosom adalah hasil mutasi dari 1 kromosom dari Generasi ke-1

Untuk Generasi ke-2, 5 kromosom adalah 5 kromosom terbaik dari Generasi ke-1, 6 kromosom adalah hasil kawin silang dari 2 kromosom dari Generasi ke-1, dan 1 kromosom adalah hasil mutasi dari 1 kromosom dari Generasi ke-1. Kromosom 1 adalah Kromosom 12 dari Generasi ke-1, Kromosom 2 adalah Kromosom 5 dari Generasi ke-1, Kromosom 3 adalah Kromosom 7 dari Generasi ke-1, Kromosom 1 adalah Kromosom 11 dari Generasi ke-1, Kromosom 5 adalah Kromosom 11 dari Generasi ke-1. Kromosom 6 adalah hasil kawin silang dari Kromosom 5 dan Kromosom 12 dari Generasi ke-1, Kromosom 7 adalah hasil kawin silang dari Kromosom 7 dan Kromosom 11 dari Generasi ke-1, Kromosom 8 adalah hasil kawin silang dari Kromosom 2 dan Kromosom 10 dari Generasi ke-1, Kromosom 9 adalah hasil kawin silang dari Kromosom 6 dan Kromosom 9 dari Generasi ke-1, Kromosom 10 adalah hasil kawin silang dari Kromosom 7 dan Kromosom 12 dari Generasi ke-1, Kromosom 11 adalah hasil kawin silang dari Kromosom 11 dan Kromosom 12 dari Generasi ke-1. Kromosom 12 adalah hasil mutasi dari Kromosom 12 dari Generasi ke-1.

# Analisis Algoritma *Hybrid Genetic*

- Berdasarkan nilai kelayakan untuk kromosom-kromosom pada Generasi ke-2 yang ditampilkan pada Tabel 3, 5 kromosom terbaik akan diambil untuk menjadi bagian dari Generasi ke-3
- Ke-5 kromosom yang terpilih adalah Kromosom 1, Kromosom 12, Kromosom 2, Kromosom 3, dan Kromosom 4

2016-12-06

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

└ Analisis

└ Algoritma *Hybrid Genetic*

└ Analisis Algoritma *Hybrid Genetic*

- Berdasarkan nilai kelayakan untuk kromosom-kromosom pada Generasi ke-2 yang ditampilkan pada Tabel 3, 5 kromosom terbaik akan diambil untuk menjadi bagian dari Generasi ke-3
- Ke-5 kromosom yang terpilih adalah Kromosom 1, Kromosom 12, Kromosom 2, Kromosom 3, dan Kromosom 4

Berdasarkan nilai kelayakan untuk kromosom-kromosom pada Generasi ke-1 yang ditampilkan pada Tabel 2, 5 kromosom terbaik akan diambil untuk menjadi bagian dari Generasi ke-2. Ke-5 kromosom yang terpilih adalah Kromosom 12, Kromosom 5, Kromosom 7, Kromosom 11, dan Kromosom 1.

# Analisis Algoritma Hybrid Genetic

Nomor Kromosom	Nilai Kelayakan
1	0,5556
2	0,4444
3	0,3889
4	0,3889
5	0,3333
6	0,2778
7	0,3889
8	0,0833
9	0,1944
10	0,2778
11	0,0833
12	0,5

Table 3: Tabel nilai kelayakan untuk kromosom-kromsوم pada Generasi ke-2

2016-12-06

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

- Analisis
  - Algoritma Hybrid Genetic
    - Analisis Algoritma Hybrid Genetic

Analisis Algoritma Hybrid Genetic

Nomor Kromosom	Nilai Kelayakan
1	0.5556
2	0.4444
3	0.3889
4	0.3889
5	0.3333
6	0.2778
7	0.3889
8	0.0833
9	0.1944
10	0.2778
11	0.0833
12	0.5

Table 3: Tabel nilai kelayakan untuk kromosom-kromsوم pada Generasi ke-2



# Analisis Algoritma *Hybrid Genetic*

- Proses yang sama diulang untuk menghasilkan Generasi ke-3 dan generasi-generasi berikutnya, hingga algoritma genetik dapat menemukan solusi dari teka-teki Calcudoku tersebut.
- Solusi ditemukan jika berhasil menemukan kromosom yang nilai kelayakannya 1.

2016-12-06

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

### Analisis

#### Algoritma *Hybrid Genetic*

#### Analisis Algoritma *Hybrid Genetic*

- Proses yang sama diulang untuk menghasilkan Generasi ke-3 dan generasi-generasi berikutnya, hingga algoritma genetik dapat menemukan solusi dari teka-teki Calcudoku tersebut.
- Solusi ditemukan jika berhasil menemukan kromosom yang nilai kelayakannya 1.

Proses yang sama diulang untuk menghasilkan Generasi ke-4 dan generasi-generasi berikutnya, hingga algoritma genetik dapat menemukan solusi dari teka-teki Calcudoku tersebut. Solusi ditemukan jika berhasil menemukan kromosom yang nilai kelayakannya 1.

## 2016-12-06

## -Daftar Pustaka

## -Daftar Pustaka



2016-12-06

└─ Thank You

—Terima Kasih

- Ada pertanyaan?