

IMPLEMENTASI ALGORITMA STOCHASTIC HILL CLIMBING PADA PERMAINAN MASTERMIND

Ruby Vidian Hartanto, Joko Purwadi, Gunawan Santosa
Program Studi Teknik Informatika Fakultas Teknik
Universitas Kristen Duta Wacana Yogyakarta
Email: jokop@ukdw.ac.id, gunawan@ukdw.ac.id

Abstrak:

Artikel ini membahas tentang implementasi algoritma Stochastic Hill Climbing (SHC) pada permainan Mastermind. Mastermind adalah salah satu jenis permainan papan yang dimainkan oleh dua orang pemain, dimana satu pemain berperan menyusun sebuah kombinasi 4 warna dari 6 buah warna yang tersedia. Sedangkan pemain lainnya bertugas menebak kombinasi warna yang dipilih oleh pemain pertama. Pemain pertama akan memberikan feedback kepada pemain kedua berdasarkan tebakan pemain kedua. Permasalahan yang dihadapi adalah bagaimana mencari langkah selanjutnya untuk memaksimalkan kemenangan.

Sistem yang dibangun untuk pencarian solusi menggunakan pendekatan algoritma Stochastic Hill Climbing. Pencarian dimulai dengan komputer memasukkan tebakan awal yang disebut CFG pada langkah pertama. Kemudian sistem mendapatkan feedback berdasarkan tebakan tersebut. Dari feedback tersebut sistem menentukan Tebakan Potensial yang akan digunakan untuk langkah berikutnya. Tebakan Potensial tersebut juga akan mendapatkan feedback, kemudian nilai heuristik dari CFG dan Tebakana Potensial akan dibandingkan, jika nilai heuristik CFG lebih kecil dari Tebakan Potensial, maka Tebakan Potensial tersebut akan dianggap sebagai CFG baru. Sebaliknya, jika nilai heuristik CFG lebih baik dari Tebakan Potensial, tentukan Tebakan Potensial berdasar CFG yang lama. Proses akan berlanjut hingga sistem mampu menebak Pola Rahasia.

Kata Kunci : heuristik, algoritma Stochastic Hill Climbing, Mastermind, kecerdasan buatan.

1. Pendahuluan

Game-game yang dibuat sekarang ini sebagian besar sudah mengimplementasikan AI (*Artificial Intelligence*) atau kecerdasan buatan. Dengan penerapan AI ini memungkinkan para pengguna sistem dapat bermain sendiri melawan sistem komputer yang mengimplementasikan

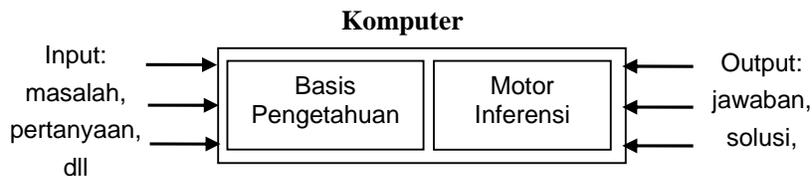
AI tanpa perlu harus menunggu orang lain. Salah satu jenis *game* yang menggunakan AI adalah *Mastermind*. Pada penelitian ini, diterapkan algoritma *Stochastic Hill Climbing* untuk sebagai cara komputer untuk menemukan solusi dari permasalahan permainan *mastermind*.

2. Landasan Teori

2.1 Kecerdasan Buatan (*Artificial Intelligence*)

Artificial Intelligence adalah sistem yang mempelajari konsep, dapat berpikir dan menarik kesimpulan dari suatu masalah, dapat mengerti bahasa dan memahami pemandangan visual, serta sistem yang melakukan tugas-tugas lain yang membutuhkan kecerdasan manusia (Patterson, D. W., 1990, halaman 1).

Sistem yang menggunakan kecerdasan buatan, memiliki input berupa masalah dan menghasilkan output berupa solusi dari masalah yang diinputkan. Untuk dapat menghasilkan solusi, sistem harus dilengkapi dengan sekumpulan pengetahuan yang ada pada basis pengetahuan, selain itu sistem juga harus memiliki motor inferensi (*inference engine*) agar mampu mengambil keputusan berdasarkan fakta atau pengetahuan (Gambar 1).



Gambar 1. Penerapan Konsep Kecerdasan Buatan di Komputer

Dikutip dari: Kusumadewi, S.,2003, halaman 3.

2.2 Algoritma Stochastic Hill Climbing

Algoritma SHC merupakan pengembangan dari algoritma *Hill Climbing*. *Stochastic Hill Climbing* memilih nilai yang digunakan secara acak berdasarkan nilai probabilitas yang muncul. Nilai probabilitas yang digunakan untuk memilih simpul berikutnya bisa berbeda berdasarkan kecuraman dari pergerakan nilai yang menuju keatas (Russell, S., Norvig, P.,1995, halaman 111).

Pada permainan *Mastermind*, berikut adalah urutan dari algoritma SHC (Temporel, A., Kovacs, T., 2003).

- Tentukan tebakan pertama, tebakan ini disebut *Current Favourite Guess*(CFG).
- Dari CFG yang telah kita masukkan, ditentukan kode baru lagi dengan metode sebagai berikut :

Langkah 1:

Jumlah warna kolom yang tetap digunakan dari tebakan sebelumnya/CFG bergantung pada angka pertama dalam kurung nilai(nilai tebakan yang tepat pada posisi yang tepat). Misalkan *feedback* yang diterima: [1,2], maka jumlah kolom yang tetap digunakan hanya 1 saja.

Langkah 2:

Kolom yang dipindahkan bergantung pada angka ke dua dalam kurung nilai(nilai tebakan yang tepat tetapi pada posisi yang salah). Misalkan *feedback* yang diterima: [1,2], maka jumlah kolom yang dipindah lokasinya adalah 2.

Langkah 3:

Kolom terakhir yang belum dioperasikan diberi nilai baru yang tidak muncul pada kolom-kolom sebelumnya. Pemberian warna pada kolom yang masih kosong dilakukan secara acak.

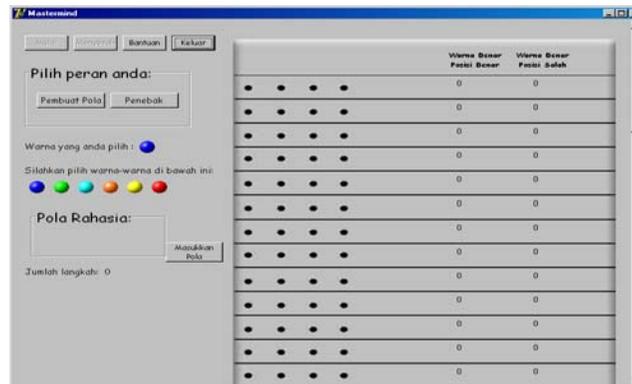
- Jika kode baru tidak konsisten terhadap semua tebakan sebelumnya, kembali ke langkah 2. tetapi jika kode baru telah konsisten, gunakan kode baru sebagai tebakan baru.
- Jika tebakan menghasilkan nilai [0,0], hilangkan semua warna yang ada pada tebakan tersebut. Tentukan kombinasi warna yang baru yang konsisten dengan semua nilai tebakan sebelumnya. Buat kombinasi baru ini sebagai CFG baru dan masukkan ke dalam kolom tebakan.
- Jika nilai tebakan yang dimasukkan sama dengan atau lebih baik dari nilai CFG menurut *heuristik* pada tabel kemungkinan *feedback*, tebakan dianggap sebagai CFG baru dan tentukan nilai baru sebagai nilai yang terbaik.
- Jika tebakan belum menghasilkan *feedback* bernilai [4, 0] ulangi langkah ke-3. Untuk nilai Heuristik *feedback* dapat dilihat pada tabel 1.
- Selesai

Tabel 1. Nilai Heuristik Feedback

| No. | Feedback |
|-----|----------|
| 0 | [0, 0] |
| 1 | [0, 1] |
| 2 | [1, 0] |
| 3 | [0, 2] |
| 4 | [1, 1] |
| 5 | [2, 0] |
| 6 | [0, 3] |
| 7 | [1, 2] |
| 8 | [1, 2] |
| 9 | [3, 0] |
| 10 | [0, 4] |
| 11 | [1, 3] |
| 12 | [2, 2] |
| 13 | [4, 0] |

3. Hasil Implementasi

Pada bagian ini disajikan hasil implementasi algoritma SHC untuk menyelesaikan permainan Mastermind. Tampilan utama dari sistem ini adalah:



Gambar 2. Form Utama Permainan Mastermind

Gambar 2 menunjukkan tampilan awal Form Utama Permainan Mastermind. Permainan dimulai melalui pemilihan peran oleh *user*, apakah sebagai Penebak dengan menekan tombol **Penebak**, atau sebagai Pembuat Pola dengan menekan tombol **Pembuat Pola**. Ketika *user* memilih peran sebagai penebak, sistem menentukan Pola Rahasia secara random. Pola Rahasia terletak di *GroupBox* Pola Rahasia dengan kondisi tidak terlihat. Pola

Rahasia akan muncul dengan 3 kondisi, yang pertama jika *user* berhasil menebak Pola Rahasia. Yang kedua, jika *user* menekan tombol Menyerah, dan membiarkan sistem memunculkan Pola Rahasia. Kondisi terakhir, jika *user* kalah, maka Pola Rahasia akan muncul.

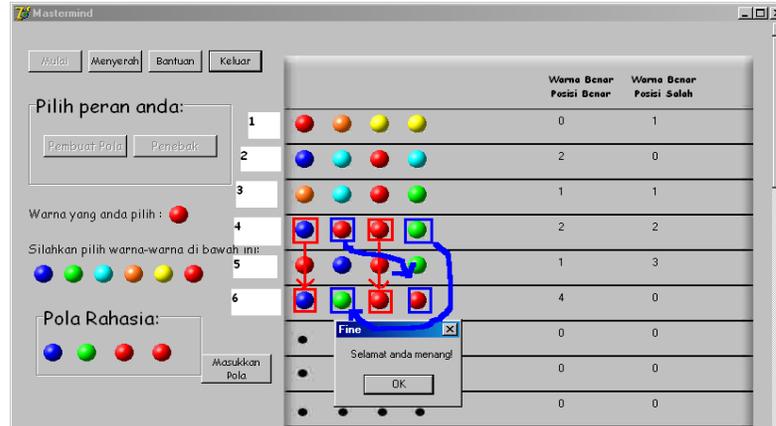
Sistem telah menyediakan 6 warna yang digunakan dalam permainan. Warna yang dipilih *user* nampak pada baris warna yang dipilih atau di atas barisan warna. Setelah *user* memilih warna yang akan digunakan sebagai tebakan, berikutnya *user* meng-klik pada kolom tebakan yang telah disediakan. Kolom tebakan terisi sesuai warna yang telah kita pilih, sesuai dengan baris warna pilihan. Jika *user* telah mengisi semua kolom tebakan, berikutnya *user* mengklik tombol **OK** yang terletak pada sebelah kolom terakhir. Berdasarkan tebakan *user*, maka sistem akan memberikan *feedback*, sebagai acuan untuk *user* dalam memilih kombinasi pada langkah berikutnya. Hingga langkah ke 20 *user* belum berhasil menebak Pola Rahasia, maka *user* dinyatakan kalah, jika sebelum langkah ke-20 *user* berhasil menebak Pola Rahasia maka *user* dinyatakan menang. Ketika *user* meng-klik tombol **Menyerah**, menandakan bahwa *user* telah menyerah dalam menebak Pola Rahasia, sehingga *user* dinyatakan kalah. Untuk kembali bermain, klik tombol Mulai yang terletak di bagian atas.

Saat *user* berperan sebagai **Pembuat Pola**, *user* wajib menentukan pola yang akan ditebak oleh sistem. Berikutnya sistem akan menebak pola dengan menggunakan metode *Stochastic Hill Climbing*. Jika sampai langkah maksimal sistem tidak berhasil menemukan Pola Rahasia, maka *user* keluar sebagai pemenang. Namun, sebelum langkah maksimal sistem berhasil menebak Pola Rahasia, maka sistemlah yang keluar sebagai pemenang.

Dalam sistem yang dikembangkan, proses algoritma SHC dibuat dalam beberapa prosedur. Proses diawali dengan prosedur penentuan Tebakan Awal (CFG), dilanjutkan prosedur Pengecekan Heuristic, prosedur penentuan Tebakan Potensial dan prosedur penghapusan warna. Prosedur-prosedur tersebut akan bekerja saat *user* selesai menentukan Pola Rahasia dan menekan tombol **Masukkan Pola**.

4. Analisis Hasil Penelitian

Pada sub bab ini akan diuraikan analisis dari implementasi algoritma SHC untuk menyelesaikan permainan Mastermind.



Gambar 3. Penyelesaian Permainan Mastermind

Langkah 1, sistem memasukkan tebakan pertama sebagai CFG dengan kombinasi **Merah-Oranye-Kuning-Kuning**, *feedback* langkah pertama adalah **[0,1]**, nilai heuristik **1**. **Langkah 2**, sistem memasukkan tebakan potensial dengan kombinasi **Biru-Biru Muda-Merah-Biru Muda**, dengan warna Merah diubah posisinya ke kolom 3. *Feedback* langkah ke-2 adalah **[2,0]**, nilai heuristik **5**. Karena nilai heuristik langkah 2 lebih besar dari langkah 1, maka kombinasi warna langkah 2 dianggap sebagai CFG. **Langkah 3**, sistem memasukkan tebakan potensial dengan kombinasi **Oranye-Biru Muda-Merah-Hijau**, dengan warna Merah dan Biru Muda pertama tetap pada posisinya. *Feedback* langkah ke-3 adalah **[1,1]**, nilai heuristik **4**. Karena nilai heuristik langkah 3 lebih kecil dari langkah 2, maka CFG tidak berubah.

Langkah 4, sistem memasukkan tebakan potensial dengan kombinasi **Biru-Merah-Merah-Hijau**, dengan warna Merah dan Biru tetap pada posisinya. *Feedback* langkah 4 **[2,2]**, nilai heuristik **12**. Karena nilai heuristik langkah 4 lebih besar dari langkah 3, maka kombinasi langkah 4 dianggap sebagai CFG. **Langkah 5**, sistem memasukkan tebakan potensial **Merah-Biru-Merah-Hijau**, dengan warna Hijau dan Merah ke-2 tetap pada posisinya, warna Biru digeser ke kolom 2 dan warna Merah pertama di geser ke kolom 1. *Feedback* langkah 5 adalah **[1,3]**, nilai heuristik **11**. Karena nilai heuristik langkah 5 lebih kecil dari langkah 4, maka CFG tidak berubah. **Langkah 6**, sistem memasukkan tebakan potensial dengan kombinasi **Biru-Hijau-Merah-Merah**, dengan warna Merah ke-2 dan warna biru pada langkah 4 tetap pada posisinya, warna Merah pertama pindah ke kolom 4 dan warna Hijau pindah ke kolom 2. *Feedback* langkah 6 adalah **[4,0]**, nilai heuristik **13**. Karena Pola Rahasia telah berhasil ditebak, maka permainan berakhir, sistem dinyatakan sebagai pemenang. Hal ini menunjukkan bahwa algoritma SHC mampu menyelesaikan permainan Mastermind.

Namun ada kalanya algoritma ini tidak mampu menyelesaikan permainan sampai langkah maksimal yang telah ditentukan. Pada beberapa kasus seperti yang tertera pada Tabel 2 terlihat bahwa SHC menyelesaikan permainan hingga 49 langkah, padahal jumlah maksimal tebakan adalah 20 langkah. Hal ini disebabkan fungsi acak yang digunakan untuk menentukan warna yang akan mengisi kolom yang masih kosong.

Tabel 2. Penyelesaian Permainan Lebih Dari 30 Langkah

| No Ujicoba | Pola Rahasia | Jumlah Langkah |
|------------|----------------------------|----------------|
| 1 | Biru-Hijau-Kuning-Merah | 32 |
| 2 | Biru-Hijau-Oranye-Kuning | 31 |
| 3 | Hijau-BiruMuda-Biru-Oranye | 49 |
| 4 | Hijau-Hijau-Merah-Oranye | 35 |
| 5 | Biru-Merah-Kuning-BiruMuda | 32 |

Selain itu juga dilakukan analisa banyaknya jenis warna untuk kombinasi Pola Rahasia terhadap jumlah langkah dalam menyelesaikan permainan Mastermind. Mengacu pada Tabel 3, semakin banyak jenis warna yang digunakan, jumlah langkah yang digunakan untuk menyelesaikan permainan. juga relatif lebih banyak.

Tabel 3. Hasil perhitungan nilai $g(n)$, $h(n)$, $f(n)$ dan $c(n)$ untuk setiap cabang Kesiman dan Tonja

| Banyak Jenis Warna | Jumlah Percobaan | Rata-rata Jumlah Langkah |
|--------------------|------------------|--------------------------|
| 1 | 20 | 5,5 |
| 2 | 20 | 10,2 |
| 3 | 20 | 16,6 |
| 4 | 20 | 23,6 |

5. Kesimpulan

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan beberapa hal mengenai Algoritma Stochastic Hill Climbing pada permainan Mastermind yaitu :

1. Metode Stochastic Hill Climbing dengan fungsi heuristic dapat digunakan untuk menyelesaikan permainan Mastermind.
2. Penerapan Metode Stochastic Hill Climbing mampu bekerja cukup baik. Namun pada beberapa ujicoba diperlukan langkah penyelesaian yang lebih banyak dari batas langkah maksimal untuk menyelesaikan permainan Mastermind.
3. Berdasarkan analisis, semakin banyak jenis warna yang digunakan sebagai Pola Rahasia, lebih banyak langkah yang diperlukan untuk menyelesaikan permainan Mastemind. Semakin sedikit jumlah warna yang dikombinasikan, jumlah langkah yang diperlukan untuk menyelesaikan permainan relatif lebih sedikit.

Daftar Pustaka

- [1] Faris , F. (April 15, 2007). Mastermind game. Diambil dari <http://www.planet-source-code.com/vb/scripts/ShowZip.asp?lngWld=7&lngCodeId=925&strZipAccessCode=tp%2FM9250122>,

- [2] Kusumadewi, S. (2003). Artificial intelligence: Teknik dan penerapannya. Graha ilmu, Yogyakarta.
- [3] Luger, G.F., Stubblefield, W.A. (1989). Artificial intelligence and the design of expert system. The Benjamin, Cummings.
- [4] Nelson, T. (March 9, 2000). Break the hidden code. Diambil dari <http://www.tnelson.demon.co.uk/mastermind/history.html>,
- [5] Patterson, D. W. (1990). Introduction to artificial intelligence and expert system. Prentice hall, Inc.
- [6] Pranata, A. (2002). Pemrograman borland delphi 6 edisi 4, Penerbit Andi, Yogyakarta.
- [7] Russell, S., Norvig, P. (1995). Artificial intelligence: A modern approach. Prentice Hall, Inc.
- [8] Temporel, A., Kovacs, T. (September 2003). A heuristic hill climbing algorithm for mastermind. *Proceedings of the 2003 UK Workshop on Computational Intelligence (UKCI-03)*. Diambil dari <http://www.cs.bris.ac.uk/Publications/Papers/2000067.pdf>.
- [9] Wahana Komputer. (2003). Pemrograman borland delphi 6.0. Penerbit Andi, Yogyakarta.