

# Penerapan Teknik *Greedy* untuk Menentukan Perpindahan Seminimal Mungkin pada Permainan *Route to Sprout*

Parel Wellman Hutahaean (Times New Roman, 12pt, Bold)

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung  
Jalan Ganesha 10 Bandung  
e-mail: if17138@students.if.itb.ac.id

## ABSTRAK

Algoritma *Greedy* merupakan salah satu cara pemecahan masalah yang lebih mangkus daripada algoritma *Brute Force*. Algoritma ini telah menerapkan prinsip optimisasi untuk menyelesaikan masalah. Namun optimisasi dengan cara *Greedy* tidak menjamin hasil yang terbaik. Tetapi untuk strategi solusi yang sederhana dan cepat terpikirkan, algoritma ini tidak rugi untuk digunakan.

Algoritma *Greedy* menyelesaikan masalah langkah per langkah. Pada setiap langkah perlu diputuskan langkah terbaik berikutnya sehingga diharapkan setiap langkah yang ditempuh bisa menjadi solusi yang optimal untuk masalah keseluruhan.

Permainan *Route to Sprout* adalah permainan yang menuntut sisi fleksibilitas dari otak manusia. Penilaian dari permainan ini dilihat berdasarkan jumlah gerakan seminimal mungkin dalam memindahkan benih ke lubangnya. Di beberapa posisi pada jalur terdapat serangga yang menghalangi pemain dari asalnya ke lubang. Perlu menyingkirkan serangga ke posisi lain agar pemain dapat melanjutkan gerakannya.

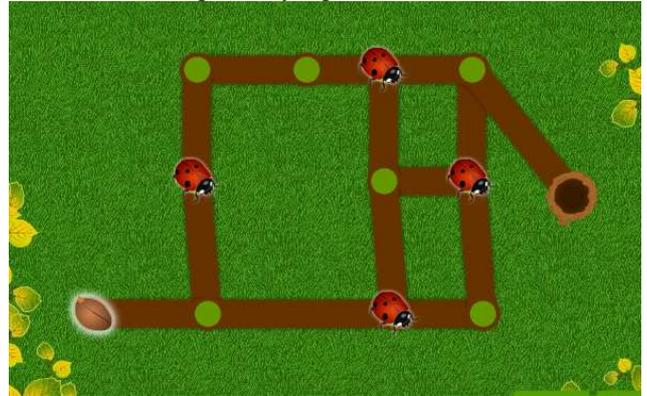
Algoritma *Greedy* dapat menjadi alternatif solusi, meskipun dijamin bukan solusi yang baik. Makalah ini dibuat sebagai rintisan bagi solusi lain yang lebih baik.

**Kata kunci:** Benih, lubang, serangga, *Greedy*, graf.

## 1. PENDAHULUAN

Permainan *Route to Sprout* diperkenalkan oleh sebuah situs berbayar [www.lumosity.com](http://www.lumosity.com). Bersama jenis permainan lainnya, permainan sederhana ini memang dirancang untuk meningkatkan kemampuan otak dalam berbagai aspek; fleksibilitas, kecepatan, memori, pemecahan masalah, dan perhatian. Situs ini juga menampilkan tips-tips untuk menjaga kebugaran otak. Sebagian permainan dapat dicoba tanpa harus mendaftar

jadi anggota (situs ini mewajibkan tarif berkala untuk bisa mencoba seluruh produk yang ditawarkan).



Gambar 1. Skrinshot Permainan *Route to Sprout*

Tujuan dari permainan ini adalah memindahkan benih dari simpul asal ke lubang. Lubang bisa lebih dari satu namun kita hanya memasukkan benih ke salah satu lubang.

Gambar 1 adalah skrinshot dari permainan sederhana ini. Ada serangga yang digambarkan dengan kepik, benih yang digambarkan dengan biji kopi, simpul-simpul yang bisa ditempati objek benih dan serangga, dan simpul khusus yaitu lubang. Masing-masing simpul dihubungkan oleh sisi yang digambarkan dengan jalan tanah.

Benih harus dipindahkan dari satu simpul ke simpul lainnya untuk dapat menuju lubang. Namun ada simpul yang ditempati oleh serangga dan benih tidak bisa dipindahkan ke simpul tersebut sampai serangga yang ada dipindahkan ke simpul lainnya yang kosong. Sisi akan diberi tanda hijau apabila perpindahan objek dari satu simpul ke simpul lain melalui sisi tersebut bisa dilakukan (artinya simpul lain itu kosong) dan tanda merah apabila tidak bisa (simpul lain itu berisi objek lain).

Perpindahan dari satu simpul ke simpul lain didefinisikan sebagai langkah. Permainan akan mencatat berapa langkah yang ditempuh pemain untuk menyelesaikannya. Semakin banyak langkah yang dilakukan, nilai akan semakin kecil. Langkah-langkah yang dihitung yaitu dari pemindahan benih dan serangga-

serangga. Jadi, tugas kita adalah mencari langkah yang paling sedikit yang bisa ditempuh.

## 2. METODE PENCARIAN SOLUSI

### 2.1 Algoritma Greedy

Algoritma *Greedy* merupakan metode yang sangat populer dalam hal optimasi. Prinsip algoritma ini adalah lakukan apa yang saat ini paling baik dilakukan. Pendekatan yang digunakan dalam algoritma *Greedy* adalah membuat pilihan yang tampaknya memberikan perolehan terbaik (optimum lokal) pada setiap langkah dengan harapan bahwa sisanya mengarah ke optimum global.

Penting membuat elemen-elemen untuk memecahkan solusi *Greedy*, yaitu sebagai berikut:

1. Himpunan kandidat yang berisi elemen pembentuk solusi
2. Himpunan solusi yang berisi kandidat yang terpilih sebagai solusi persoalan
3. Fungsi seleksi sebagai fungsi yang memungkinkan mencapai solusi paling optimal pada setiap langkah memilih kandidat.
4. Fungsi kelayakan yang memastikan kandidat yang dipilih memberikan solusi yang layak.
5. Fungsi obyektif yaitu fungsi yang memaksimalkan atau meminimumkan nilai solusi.

### 2.2 Pemecahan Permainan

Dalam memecahkan persoalan permainan ini, saya membaginya ke dalam 2 fase, antara lain:

1. Fase mencari jalur tersingkat
2. Fase memindahkan benih

Fase *jalur tersingkat* merupakan fase dalam memilih jalur tersingkat yang bisa ditempuh dari simpul asal ke lubang. Objek yang diperhatikan adalah benih semata-mata, tidak melihat serangga yang menghalangi langkahnya.

Pada gambar 1, jalur tersingkat untuk benih dari simpul asal ke lubang ada 6 langkah. Namun di ada banyak alternatif jalur yang dapat ditempuh oleh pemain. Oleh sebab itu, diperlukan fase memindahkan benih.

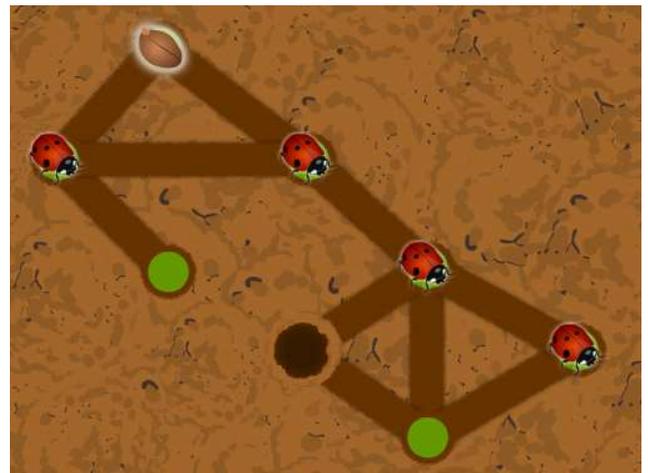
Fase *memindahkan benih* adalah fase di mana untuk melangkah perlu dipikirkan ke mana selanjutnya perpindahan baik itu benih maupun serangga pada satu posisi benih tersebut.

### 2.3 Cara Kerja

Pada fase pertama, mencari jalur tersingkat, kita hanya perlu melihat jarak terdekat dari simpul asal ke lubang. Fase ini merupakan fase prioritas utama sebelum melanjutkan ke fase kedua.

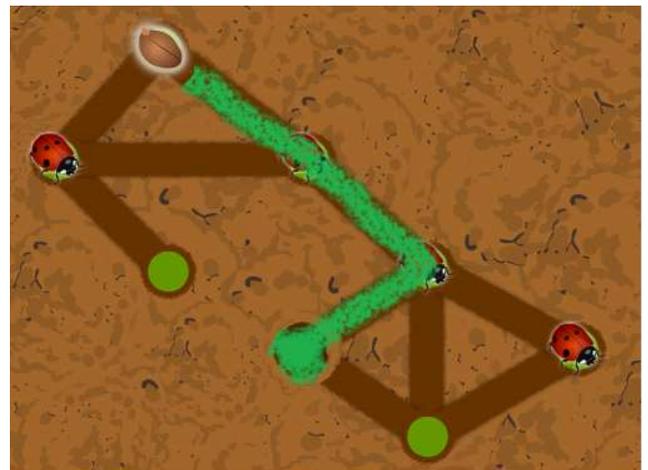
Prinsip algoritma *Greedy* yang dipakai adalah melakukan langkah optimum pada tahap awal permainan yaitu dengan menentukan jalur (atau jalur-jalur) yang pasti harus dilewati oleh benih, dengan berharap nantinya pada fase kedua akan dicapai hasil yang optimal bagaimanapun konsekuensinya. Konsekuensi di sini maksudnya adalah memindahkan serangga yang menghalangi sedemikian rupa sehingga simpul berikutnya yang harus dilewati benih (sesuai jalur yang telah dipilih) kosong.

Sebagai contoh graf yang lebih kompleks dari gambar 1 adalah sebagai berikut:



Gambar 2. Screenshot Graf *Route to Sprout*

Dari graf pada gambar 2 kita temukan rute paling singkat seperti di bawah ini:



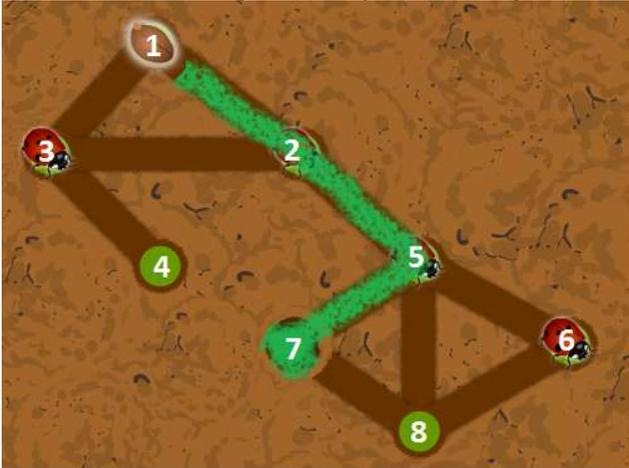
Gambar 3. Screenshot Jalur Tersingkat dari Asal ke Lubang

Penentuan jalur tersingkat seperti ini menggunakan algoritma *Dijkstra*. Algoritma *Dijkstra* dipakai untuk mencari jarak terdekat (*shortest path*) ke seluruh simpul yang ada. Jarak terdekat dari suatu simpul ke simpul lain

awalnya dilihat dari simpul tetangga, kemudian dihitung jarak simpul tetangga ke simpul lain tersebut.

Jika hanya ada satu jalur menuju lubang dalam permainan ini, maka jalur itu dipastikan yang dilewati oleh benih. Sedangkan jika ada beberapa alternatif jalur, kita perlu memikirkan perpindahan benih menurut jalur yang paling sederhana dengan melihat jumlah serangga pada masing-masing jalur alternatif. Jalur dengan jumlah serangga paling sedikit adalah jalur yang paling baik menurut asumsi. Sementara apabila ada jalur yang sama-sama paling baik, terserah kita mau mengikuti jalur yang mana.

Detail solusi untuk gambar 2 dijabarkan sebagai berikut:



**Gambar 3. Skripsut Graf dengan Simpul yang dinomori**

1. Pindahkan serangga pada simpul 3 ke 4 (karena menghalangi benih melalui jalur)
2. Pindahkan serangga pada simpul 2 ke 3
3. Pindahkan benih ke simpul 2
4. Pindahkan serangga pada simpul 5 ke 8
5. Pindahkan benih ke simpul 5
6. Pindahkan benih ke simpul 7

### **3. KESIMPULAN**

Dalam memecahkan masalah pada permainan *Route to Sprout*, algoritma *Greedy* cukup baik digunakan untuk memperoleh nilai tinggi. Namun memang tidak dapat dijamin kalau metode ini bisa memenuhi langkah yang paling optimum yang bisa dilakukan oleh pemain. Setidak-tidaknya pemain bisa menemukan jalan menuju lubang dan cukup sangkil jika dibandingkan dengan memakai metode *Brute Force*.

### **REFERENSI**

- [1] Rinaldi Munir, "Diktat Kuliah IF3051 Strategi Algoritma", 2008.
- [2] Anany Levitin, "Introduction to The Design and Analysis of Algorithm", Addison Wesley, 2003.