

# IMPLEMENTASI ALGORITMA *GREEDY* PADA PERMAINAN CONGKLAK

Ripandy Adha - NIM 13507115

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Jalan Ganesha nomor 10  
e-mail: [if17115@students.itb.ac.id](mailto:if17115@students.itb.ac.id), [ree\\_p\\_and\\_y@yahoo.com](mailto:ree_p_and_y@yahoo.com)

## ABSTRAK

Algoritma *Greedy* adalah algoritma yang umum digunakan dalam pemecahan masalah optimasi. Secara harfiah, kata *greedy* berarti tamak. Prinsip yang digunakan dalam algoritma *greedy* adalah "*take what you can get now*", yaitu mengambil kesempatan yang ada tanpa memikirkan konsekuensi kedepannya. Makalah ini akan membahas mengenai implementasi algoritma *greedy* dalam permainan congklak. Permainan congklak adalah permainan dengan 2 orang pemain, menggunakan biji dengan arena permainan umumnya memiliki 16 buah lubang tempat biji diletakkan, dimana 2 lubang merupakan "lubang penyimpanan" milik masing-masing pemain. Tujuan dari permainan congklak adalah mendapatkan sebanyak mungkin biji pada "lubang penyimpanan" sendiri. Penerapan algoritma *greedy* yang dapat diimplementasikan dalam permainan congklak adalah dengan mengambil langkah yang dapat memberikan biji terbanyak ke dalam "lubang penyimpanan".

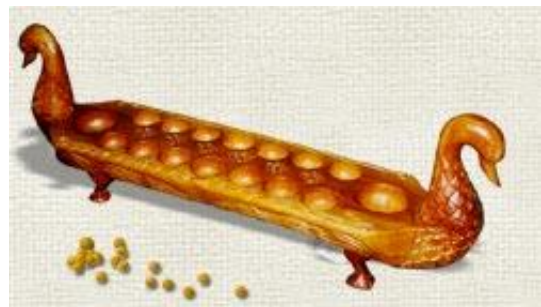
**Kata kunci:** Algoritma Greedy, Congklak.

## 1. PENDAHULUAN

### 1.1 Permainan Congklak

Permainan congklak adalah permainan tradisional Indonesia yang dimainkan oleh dua orang pemain. Permainan dilakukan dengan menggunakan 98 buah biji dan papan permainan berbentuk lonjong yang umumnya memiliki 7 buah lubang di setiap sisinya. Selain itu juga terdapat dua lubang di ujung papan, yang umumnya terletak di tengah, berukuran lebih besar, dan digunakan sebagai "lubang penyimpanan" dalam permainan. Lubang-lubang tersebut adalah tempat biji congklak dimainkan. Tujuan dari permainan congklak adalah mendapatkan sebanyak mungkin biji ke dalam lubang penyimpanan milik sendiri. Pemenang dalam permainan congklak

adalah pemain dengan jumlah biji terbanyak pada lubang penyimpanan miliknya, atau pemain yang masih dapat memainkan biji pada gilirannya, yaitu saat pemain lawan kehabisan biji di sisi papannya, dan tidak dapat melanjutkan permainan. Permainan congklak pada umumnya dimainkan dalam beberapa babak, dimana babak pertama dengan babak selanjutnya memiliki peraturan yang sedikit berbeda. Akan tetapi, dalam makalah ini hanya akan dibahas permainan congklak pada babak pertama saja.



Gambar 1. Papan Permainan Congklak

### 1.2 Algoritma *Greedy*

Algoritma *greedy* merupakan metode yang paling populer untuk memecahkan masalah optimasi. Secara harfiah, *greedy* berarti tamak atau rakus. Prinsip dari algoritma *greedy* adalah mengambil setiap kesempatan yang ada saat itu juga, tanpa memperhatikan konsekuensi kedepannya. Algoritma *greedy* membentuk solusi dari langkah demi langkah, dan pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Di setiap langkahnya algoritma *greedy* mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi kedepan. Setiap keputusan yang diambil diharapkan merupakan langkah optimum pada langkah tersebut, dikenali sebagai solusi optimum lokal, kemudian dengan setiap langkah yang ditempuh diharapkan dapat memperoleh solusi optimum di akhir proses, yaitu solusi optimum global.

Algoritma *greedy* tidak selalu memperoleh solusi optimum untuk keseluruhan permasalahan yang ditangani, dikarenakan algoritma *greedy* tidak melakukan operasi secara *exhaustive* kepada semua data, dan seringkali tidak mementingkan solusi optimum global. Akan tetapi, algoritma *greedy* merupakan solusi yang baik digunakan dikarenakan algoritma *greedy* bekerja dengan cepat dan sering memberikan perkiraan nilai optimum yang baik di setiap langkahnya. Dan tidak jarang dapat menghasilkan solusi optimum global pada suatu permasalahan dari pengambilan solusi optimum lokal di setiap langkahnya.

Elemen-elemen algoritma *greedy* dalam persoalan optimasi adalah sebagai berikut :

1. Himpunan kandidat  
Himpunan ini berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi  
Himpunan ini berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Himpunan solusi merupakan himpunan bagian dari himpunan kandidat.
3. Fungsi seleksi  
Fungsi seleksi dinyatakan sebagai predikat SELEKSI merupakan fungsi yang pada setiap langkah memilih kandidat yang paling mungkin untuk mendapatkan solusi optimal.
4. Fungsi kelayakan  
Fungsi kelayakan dinyatakan sebagai predikat LAYAK merupakan fungsi yang memeriksa apakah suatu kandidat yang dipilih dapat memberikan solusi yang layak, yaitu kandidat tersebut tidak melanggar aturan yang ada.
5. Fungsi objektif  
Fungsi objektif merupakan fungsi yang memaksimumkan atau meminimumkan nilai solusi.

## 2. METODE

### 2.1 Aturan Permainan Congklak

Permainan congklak dimainkan oleh dua orang pemain. Permainan dilakukan bergiliran secara bergantian. Permainan congklak dimainkan diatas sebuah papan lonjong yang masing-masing sisi memiliki tujuh lubang dan dua lubang penyimpanan di kedua ujung papan. Masing-masing pemain menghadap ke sisi papan dan saling berhadapan. Lubang penyimpanan tiap pemain berada di sebelah kiri pemain. Permainan dilakukan selama beberapa babak, dimana babak pertama memiliki sedikit perbedaan peraturan dengan babak-babak selanjutnya.

Untuk babak pertama, pada awal permainan, masing-masing lubang diisi tujuh buah biji, tetapi dua lubang penyimpanan dibiarkan kosong. Pemain pertama mengambil seluruh biji di lubang manapun pada sisinya, kemudian melakukan langkah searah jarum jam menelusuri lubang-lubang sambil meletakkan satu biji di tiap lubang yang dilalui dalam usaha mengisi lubang penyimpanan. Biji-biji diletakkan di setiap lubang termasuk lubang-lubang di sisi lawan dan lubang penyimpanan milik sendiri, tetapi tidak termasuk lubang penyimpanan milik lawan.

Pada saat melangkah, terdapat beberapa kasus untuk menangani langkah dimana biji terakhir berhenti :

1. Jika biji terakhir dalam melakukan langkah adalah pada lubang yang masih terdapat biji, maka ambil semua biji pada lubang tersebut, dan lakukan langkah seperti sebelumnya.
2. Jika biji terakhir jatuh di lubang yang kosong, maka tinggalkan biji terakhir di lubang tersebut, dan saat itu giliran berakhir, bergantian dengan pemain lawan.
3. Jika biji terakhir jatuh di lubang penyimpanan, maka pemain mendapat giliran sekali lagi.
4. Jika biji terakhir jatuh di lubang kosong di sisi milik sendiri, serta telah mengelilingi papan sebanyak paling sedikit satu kali, maka jika lubang seberangnya tidak kosong (lubang pada sisi lawan yang berhadapan dengan lubang tempat biji terakhir tersebut), maka ambil biji terakhir tersebut dan semua biji di lubang di seberangnya, dan simpan di lubang penyimpanan, kemudian saat itu juga giliran berakhir, bergantian dengan pemain lawan. Keadaan ini sering dikenal dengan istilah "tembak" dalam permainan.

Saat giliran pemain pertama berakhir, pemain kedua melakukan cara bermain yang sama dengan yang dilakukan pemain pertama.

Suatu babak berakhir ketika salah satu pemain kehabisan biji di sisi papan miliknya. Pemain yang terlebih dahulu kehabisan biji kalah pada babak tersebut, dan pemain lawan menjadi pemenang di babak tersebut. Pemain yang menang pada suatu babak mendapat giliran jalan pertama pada babak selanjutnya.

Babak kedua dan seterusnya memiliki peraturan khusus yang berbeda dari babak pertama, tergantung dari hasil permainan pada babak sebelumnya. Akan tetapi, makalah ini tidak akan membahas mengenai hal tersebut, karena masalah yang dibahas dibatasi pada permainan babak pertama.

Permainan berakhir jika salah satu pemain kehabisan seluruh biji nya, atau kedua pemain memutuskan untuk berhenti bermain. Pada saat akhir permainan, kedua pemain menghitung jumlah biji yang dimilikinya untuk menentukan pemenang dari permainan.

## 2.2 Pendekatan Algoritma Greedy

Algoritma *greedy* pada permainan congklak dapat diterapkan dalam mencapai tujuan permainan, yaitu mendapatkan biji terbanyak di lubang penyimpanan. Pendekatan untuk algoritma *greedy* dapat disesuaikan dengan peraturan permainan congklak, dengan menentukan prioritas pengambilan langkah akan diawali pada lubang yang paling menguntungkan.

Strategi pertama, yang paling baik untuk bisa mendapat sebanyak mungkin biji di lubang penyimpanan, adalah mendapat giliran sebanyak mungkin dengan cara mengambil langkah yang akan menjatuhkan biji terakhir pada lubang penyimpanan. Jika terdapat beberapa pilihan, maka pilihan terbaik adalah dengan memilih langkah pada lubang terdekat dari lubang penyimpanan, karena dapat dipastikan pemilihan langkah tersebut terlebih dahulu akan memberikan kesempatan yang sama saat mengambil langkah lainnya.

Jika tidak ada langkah yang memungkinkan untuk mendapatkan giliran lagi dengan cara yang telah disebutkan sebelumnya, strategi kedua adalah dengan melakukan "tembak" kepada lawan, dengan cara mencari langkah yang akan memberikan kemungkinan untuk dapat melakukan satu putaran, dan berhenti pada lubang kosong di sisi milik sendiri, dimana pada lubang yang tepat berseberangan pada sisi lain memiliki jumlah biji paling banyak, sehingga semua biji terakhir dan biji lawan dapat diletakkan seluruhnya dalam lubang penyimpanan dan menjadi milik sendiri.

Kedua strategi tersebut merupakan strategi terbaik yang dapat digunakan dalam permainan, akan tetapi ketika kedua strategi tersebut tidak dapat dilakukan, maka strategi default yang akan digunakan oleh algoritma *greedy* adalah dengan mengambil langkah dari lubang dengan biji terbanyak dan paling sedikit dapat mencapai lubang penyimpanan sebanyak satu kali, dengan harapan akan dapat terus melanjutkan langkah selama mungkin.

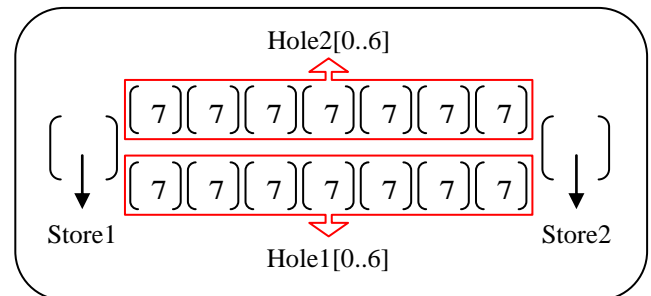
## 3. REPRESENTASI ALGORITMIK

### 3.1 Representasi Papan Congklak

Bentuk papan permainan congklak direpresentasikan dalam notasi algoritmik :

1. Hole1 : Merupakan larik bertipe integer sebagai representasi papan pada sisi pemain pertama dimana larik dengan indeks [0..6] merepresentasikan tujuh lubang papan permainan dan nilai pada larik lubang dengan index tertentu merepresentasikan jumlah biji dalam larik lubang di indeks tersebut. Indeks terurut membesar dimulai dari lubang yang paling dekat dengan lubang penyimpanan hingga yang paling jauh.

2. Hole2 : Merupakan larik bertipe integer sebagai representasi papan pada sisi pemain kedua dimana larik dengan indeks [0..6] merepresentasikan tujuh lubang papan permainan dan nilai pada larik lubang dengan index tertentu merepresentasikan jumlah biji dalam larik lubang di indeks tersebut. Indeks terurut membesar dimulai dari lubang yang paling dekat dengan lubang penyimpanan hingga yang paling jauh.
3. Store1 : Merupakan representasi lubang penyimpanan pemain pertama dengan tipe integer. Nilai integer merepresentasikan jumlah biji pada lubang penyimpanan tersebut.
4. Store2 : Merupakan representasi lubang penyimpanan pemain kedua dengan tipe integer. Nilai integer merepresentasikan jumlah biji pada lubang penyimpanan tersebut.



Gambar 2. Representasi Papan Permainan Congklak

### 3.2 Representasi Algoritma Greedy

Algoritma *greedy* dalam masalah pengambilan langkah dalam permainan congklak adalah :

1. Himpunan kandidat C, adalah semua lubang dalam permainan congklak.
2. Himpunan solusi S, adalah salah satu dari himpunan kandidat yang dapat memberikan solusi maksimal.
3. Fungsi objektif adalah mengambil langkah yang menghasilkan paling banyak biji di lubang penyimpanan dalam satu giliran.
4. Fungsi kelayakan adalah solusi bukan merupakan lubang yang tidak ada bijinya.
5. Fungsi seleksi adalah fungsi yang menentukan lubang manakah yang menjadi solusi dari algoritma *greedy* dan yang paling mungkin dapat memberikan solusi paling optimal.

Dalam masalah ini, fungsi seleksi dapat direalisasikan dengan menggunakan strategi yang telah dibahas pada bab sebelumnya. Disini, Algoritma *greedy* akan memilih langkah tergantung kepada tingkat baiknya strategi yang digunakan. Fungsi seleksi akan selalu menggunakan strategi terbaik yang memungkinkan untuk dilakukan.

Pseudo-code untuk fungsi seleksi dapat dituliskan seperti berikut :

```
function greedyCongklak (input
Hole[0..6] : array of integer) ->
integer
{Merupakan fungsi yang mengembalikan
nilai index dari larik lubang yang
dipilih sebagai langkah yang tepat}

kamus lokal

function isStg1Avail (input
Hole[0..6] : array of integer) ->
boolean
{merupakan fungsi yang mengembalikan
nilai true jika strategi pertama
dapat dilakukan}

function isStg2Avail(input Hole[0..6]
: array of integer) -> boolean
{merupakan fungsi yang mengembalikan
nilai true jika strategi kedua dapat
dilakukan}

function getIndexStg1(input
Hole[0..6] : array of integer) ->
integer
{mengembalikan index larik lubang
terendah yang mungkin untuk strategi
pertama}

function getIndexStg2(input
Hole[0..6] : array of integer) ->
integer
{mengembalikan index larik lubang
mungkin untuk strategi kedua}

function getMaks(input Hole[0..6] :
array of integer) -> integer
{merupakan fungsi yang akan
mengembalikan index larik lubang yang
memiliki biji paling banyak}

idx : integer

Algoritma

if (isStg1Avail(Hole)= true) then
  idx <- getIndexStg1(Hole)
else if (isStg2Avail(Hole)=true) then
  idx <- getIndexStg2(Hole)
else
  idx <- getMaks(Hole)
endif
-> idx
```

Fungsi *greedyCongklak* adalah fungsi yang mengembalikan nilai indeks dari larik lubang yang merupakan pilihan langkah terbaik yang dapat dilakukan. Pada awalnya fungsi ini akan memeriksa apakah langkah strategi pertama, yaitu dengan mengambil langkah yang berakhir di lubang penyimpanan, dapat dilakukan pada saat itu, dengan bantuan fungsi *isStg1Avail* sebagai pemeriksanya. Jika fungsi *isStg1Avail* mengembalikan nilai *true*, maka menandakan bahwa langkah strategi pertama dapat dilakukan, dan kemudian dengan bantuan fungsi *getIndexStg1* akan mencari indeks larik lubang yang paling dekat dengan lubang penyimpanan untuk strategi yang maksimal.

Ketika fungsi *isStg1Avail* tidak mengembalikan nilai *true*, berarti strategi pertama tidak dapat dijalankan pada langkah tersebut. Maka dari itu dilakukan pemeriksaan selanjutnya, apakah langkah strategi kedua, yaitu dengan mengambil langkah yang dapat melakukan "tembak" pada lawan dengan keuntungan maksimum, dapat dilakukan. Pemeriksaan dilakukan dengan bantuan fungsi *isStg2Avail*. Ketika fungsi *isStg2Avail* mengembalikan nilai *true*, maka menunjukkan bahwa langkah strategi kedua dapat dilakukan, dan kemudian dengan bantuan fungsi *getIndexStg2* akan mencari indeks larik lubang yang dapat melakukan "tembak" dengan keuntungan terbesar.

Apabila fungsi *isStg2Avail* juga tidak mengembalikan nilai *true* setelah melakukan pemeriksaan terhadap kedua strategi terbaik, berarti kedua strategi permainan tidak dapat dilakukan pada langkah tersebut, sehingga strategi terakhir yang dapat digunakan adalah dengan memilih langkah pada lubang dengan jumlah biji terbanyak yang dapat diambil. Fungsi *getMaks* akan membantu mencari indeks larik lubang yang memiliki jumlah biji terbanyak pada saat itu.

#### 4. KESIMPULAN

Algoritma *greedy* adalah algoritma yang sederhana, cepat, praktis, dan cukup baik dalam pemecahan masalah optimasi, walaupun hasil yang diberikan tidak selalu optimal secara keseluruhan.

Pemecahan masalah permainan congklak dapat diselesaikan dengan pendekatan algoritma *greedy* untuk mendapatkan biji sebanyak mungkin pada lubang penyimpanan milik sendiri. Pendekatan dilakukan dengan memperhatikan aturan permainan dan menyusun strategi terbaik yang dapat dilakukan sesuai dengan aturan permainan yang berlaku. Algoritma *greedy* dapat diimplementasikan dengan menerapkan strategi terbaik tersebut dalam setiap keputusan pengambilan langkah di setiap kesempatan untuk memperoleh langkah yang sebisa mungkin dapat memberikan biji terbanyak ke dalam lubang penyimpanan.

## **REFERENSI**

- [1] Munir, Rinaldi. Diktat Kuliah Strategi Algoritmik. 2007. Bandung : Program Studi Teknik Informatika ITB.
- [2] <http://www.egpat.or.id/info/conglak.html>. Diakses pada tanggal 29 Desember 2009.
- [3] <http://www.egpat.or.id/info/conglakinstructions.html#top>. Diakses pada tanggal 29 Desember 2009.