# Pencarian Solusi Optimal dalam Permainan Congklak dengan Program Dinamis

**Muchamad Surya Prasetyo**

Program Studi Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail: if15065@students.if.itb.ac.id

## ABSTRAK

**Permainan congklak adalah suatu permainan tradisional. Dimana permainan ini dibutuhkan sebuah papan yang memiliki 16 lubang dan 2 diantaranya sebagai lubang penampung. Lubang penampung berada di ujung sisi sebelah kiri dari lubang terakhir sisi pemain. Selain papan, diperlukan juga biji atau kerang sebanyak 98 buah, masing-masing lubang diisi 7 buah kecuali lubang penampung.**

**Makalah ini disusun untuk menentukan langkah yang lebih optimal setiap kali pemain mendapat giliran dengan melihat jumlah pemain mengambil biji dari lubang pada satu giliran, jumlah biji di lubang penampung dan kemungkinan pemain untuk melakukan "tembak". Tembak berarti pemain dapat mengambil biji yang berada di lubang musuh untuk disimpan ke lubang penampung.**

**Dengan melihat beberapa aspek dan harus ditinjau dari setiap lubang pemain maka digunakan algoritma Program Dinamis. Algortima ini mencari solusi optimal dengan melihat semua sisi, serta solusi bergantung pada langkah yang diambil sebelumnya, sehingga sesuai untuk menentukan solusi optimal dalam permainan congklak. Walaupun bila dilihat dari kecepatan dalam menemukan solusi lebih lama, namun solusi ini lebih optimal setelah dilakukan percobaan.**

**Untuk menyusun makalah ini dilakukan percobaan secara langsung dengan memainkan permainan congkak serta membuat algoritma pemrogramannya dari analisis permainan. Serta mencari bahan-bahan referensi melalui internet dan dengan mempelajari buku "Diktat Kuliah IF2251 Strategi Algoritmik" yang disusun oleh Ir. Rinaldi Munir, M.T.**

**Kata kunci:** *Pemrograman Dinamis, Congklak, Greedy, default*

## 1. PENDAHULUAN

### 1.1 Sejarah

Congklak ditemukan pertama kali oleh bangsa Arab atau Afrika, tergantung yang kita yakini. Lalu datang ke Asia melalui pedagang Arab dan ke Kairbia sekitar tahun 1640 melalui perdagangan budak asal Afrika.
Di Indonesia permainan congklak adalah sebuah permainan tradisional yang berasal dari pedagang asing yang digunakan untuk berdakwah pada kalangan nigrat.

### 1.2 Cara Bermain

Permainan congklak merupakan permainan dimana pemain dengan jumlah biji di lubang penampung lebih banyak adalah yang menang. Lubang penampung memiliki ukuran lebih besar dari lubang lainnya dan kosong pada awalnya, sedangkan lubang yang lain berisi 7 biji.

Setiap pemain mengambil semua biji dari salah satu lubang dari 7 lubang yang dia miliki. Lalu biji tersebut diisikan ke lubang-lubang selanjutnya menurut arah jarum jam hingga biji di tangan habis. Apabila biji di tangan habis ketika mengisi lubang penampung atau berada di lubang miliknya maka pemain dapat mendapat giliran sekali lagi. Jika pengisian terakhir di lubang penampung maka peain dapat bebas menentukan lubang mana yang selanjutnya akan dia ambil, sedangkan bila berakhir di lubang lain di sisi pemain, maka lubang tersebut yang selanjutnya akan diambil bijinya dan diisikan ke lubang lainnya. Jika pengisian terakhir berada di lubang sisi pemain dan di lubang tersebut kosong maka pemain akan dapat "menembak". "Menembak" adalah saat pemain dapat mengambil biji di lubang tersebut dan mengambil biji di lubang musuh yang berada dihadapan lubang yang kosong. Giliran pemain berakhir ketika pengisian biji terakhir berada di lubang musuh.

## 2. METODE

### 2.1 Deskripsi Pemrograman Dinamis

Program dinamis adalah sutau metode penyelesaian masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (*step*) atau tahapan (*stage*) sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan.

Program dinamis memiliki kemiripan dengan metode *greedy*, bedanya adalah dalam *greedy* kita membuat keputusan pada setiap tahap dengan cara mengambil pilihan yang paling menarik (yang memenuhi ukuran optimasi yang digunakan). Pengambilan keputusan pada setiap langkah didasarkan pada informasi lokal, dan pada setiap langkah itu kita tidak pernah membuat keputusan yang salah. Namun itu terjadi pada persoalan-persoalan tertentu saja, banyak persoalan lain yang tidak mendapatkan solusi optimal dengan menggunakan metoda *greedy*.

Hal tersebut terjadi karena metoda *greedy* tidak mempertimbangkan bagaimana langkah-langkah berikutnya apakah tepat atau tidak pada akhirnya. Sedangkan dengan program dinamis, kita dapat secara drastis mengurangi pengenumerasian rangkaian keputusan yang tidak mengarah ke solusi optimum dan melihat bagaimana langkah-langkah ke depan apakah optimum atau tidak.

### 2.2 Penemuan Solusi Optimum

Untuk menemukan solusi optimum dalam permainan congklak, dilakukan 3 tahapan sebagai berikut :

1. Melihat apakah ada kemungkinan untuk menembak
2. Mencari tujuan akhir lubang penampung
3. Mencari langkah untuk mendapat giliran terbanyak

### 2.2.1 "Tembak"

"Tembak" adalah suatu kondisi dimana pemain dapat mengambil biji yang dia miliki dan mengambil biji yang musuh miliki. Pemain dapat "menembak" hanya bila pemain mengisi biji terakhir di lubang kosong miliknya.

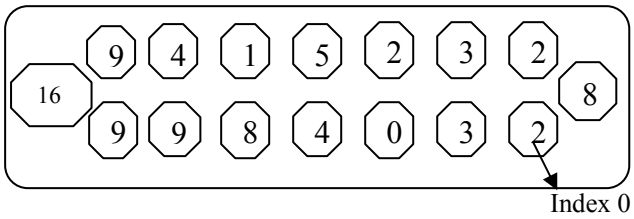
Mekanisme dalam menentukan langkah dengan melihat kesempatan untuk "menembak" :

1. Melakukan iterasi dari lubang ke-1 hingga lubang ke-7.
2. Cek bila ada yang kosong
3. Bila ada, mengembalikan index lubang.
4. Cek jumlah biji setiap lubang sebelum lubang kosong. Jumla biji harus sesuai dengan rumus

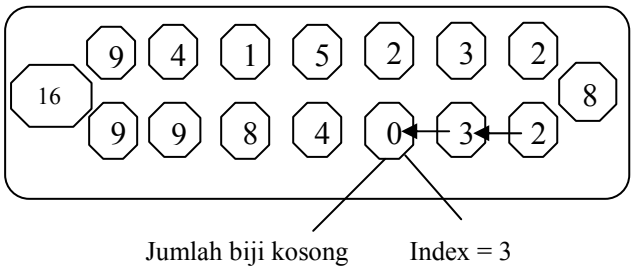
$$\Sigma biji = I_{lubang\ kosong} - I_{lubang\ cek} \quad (1)$$

5. Bila ada, cek jumlah biji dihadapan lubang kosong lalu jumlahkan dengan 1.

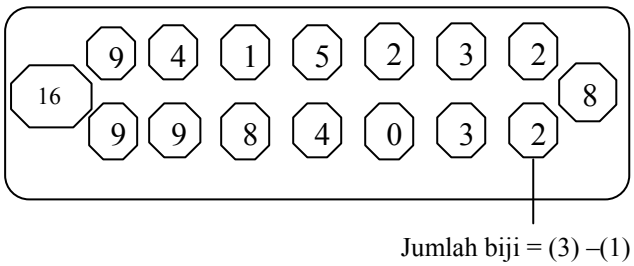
Hasil dari jumlah tersebut adalah banyaknya biji yang akan ditambahkan ke lubang penampung sebagai bobot dalam graf untuk penentuan solusi optimum.
Index dimulai dari 0 di sisi kanan bawah, menaik ke arah jarum jam.
Contoh untuk tahap diatas adalah sebagai berikut :



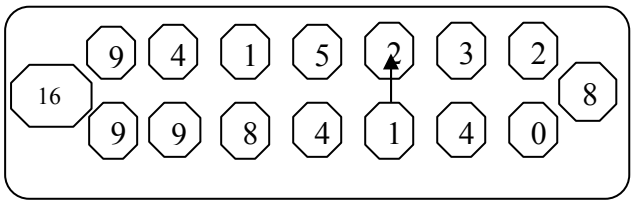

**Gambar 2.2.1.a. Sebelum melakukan pengecekan**



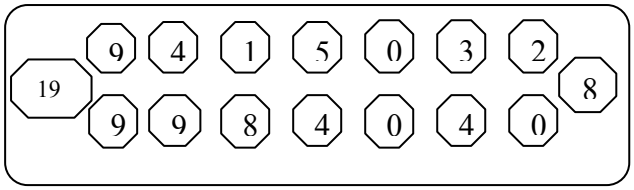

**Gambar 2.2.1.b. Pengecekan lubang kosong**




**Gambar 2.2.1.c. Pencarian lubang dengan jumlah biji sesuai rumus (1)**




**Gambar 2.2.1.d. Jika langkah nembak dipilih**



**Gambar 2.2.1.e. Hasil dari gbr 2.2.1.d**

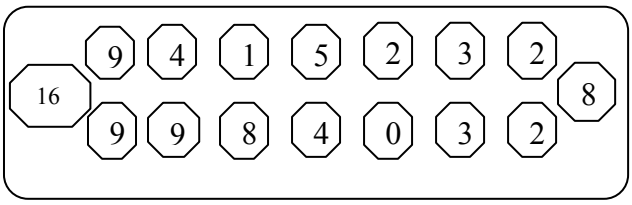
Dari contoh diatas dapat dilihat bahwa bila langkah "Tembak" diambil maka hanya akan menambahkan 3 biji pada lubang penampung.

## 2.2.2 Langkah Akhir di Lubang Penampung

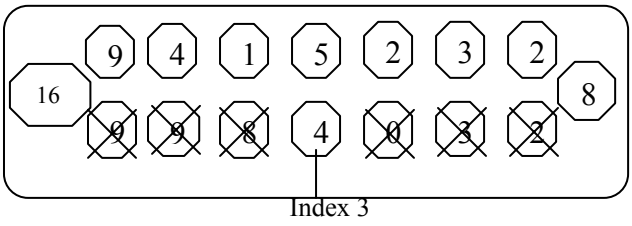
Tahap selanjutnya adalah melakukan pengecekan pada setiap lubang dimana akan berakhir bila diambil langkah pada lubang tersebut.

Mekanisme pada langkah ini terbilang cukup mudah hanya melihat jumlah biji dari setiap lubang apakah jumlahnya sama dengan 7 – index lubang atau tidak. Apabila ditulis dengan rumus, sebagai berikut :

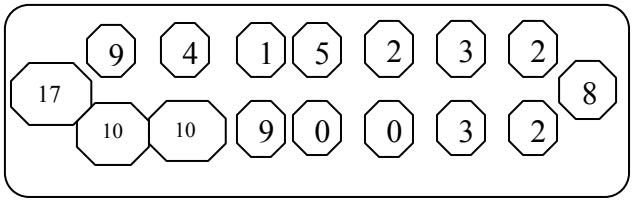
$$\Sigma biji = 7 - I_{lubang\ cek} \qquad (2)$$




**Gambar 2.2.2.a. Sebelum melakukan pengecekan**




**Gambar 2.2.2.b. Pengecekan tiap lubang**




**Gambar 2.2.2.c. Jika langkah di gbr 2.2.2.b. diambil**

Jika pemain melakukan tahap "Mencari Langkah akhir di Lubang Penampung" maka pemain dapat melakukan langkah lainnya dengan bebas menentukan lubang mana yang akan digunakan untuk langkah selanjutnya. Bisa menggunakan tahap "Tembak" atau tahap yang sama atau tahap "Mencari jumlah giliran terbanyak"

## 2.2.3 Jumlah Giliran Terbanyak

Tahap ini adalah tahap yang dapat menggabungkan 2 tahap diatas karena setelah melakukan langkah sesuai mekanisme yang akan dijelaskan nanti, pemain dapat mengambil langkah 2, dapat pula melakukan langkah 1.

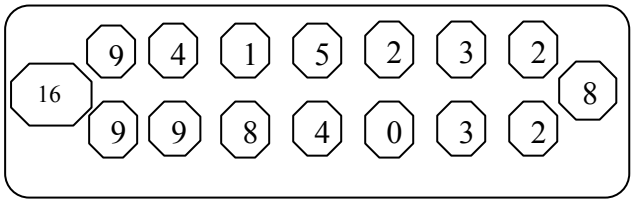
Mekanisme dalam pengambilan langkah agar mendapatkan jumlah giliran terbanyak adalah sebagai berikut :

1. Mengecek jumlah biji setiap lubang
2. Jumlah biji harus sesuai dengan rumus :

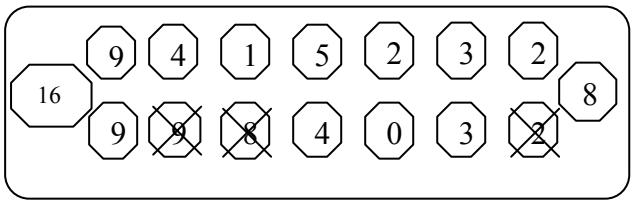
$$F(x) = \Sigma biji \leq \begin{cases} 7 - I_{lubang\ cek,} & \Sigma biji \leq 7 \\ F(x) - 15 - I_{lubang\ cek,} & \Sigma biji > 7 \end{cases} (3)$$

3. Cek langkah selanjutnya, bila berhenti di langkah selanjutnya cek lubang yang lain

Mekanisme tahap ini hampir serupa pada tahap 2. Bedanya tahap ini pemain tidak hanya menganalisis lubang yang berakhir di lubang penampung. Penjelasan lebih lengkapnya mengenai tahap-tahap diatas akan dijelaskan pada bagian analisis.




**Gambar 2.2.3.a. Sebelum melakukan pengecekan**




**Gambar 2.2.3.b. Setelah melakukan pengecekan**

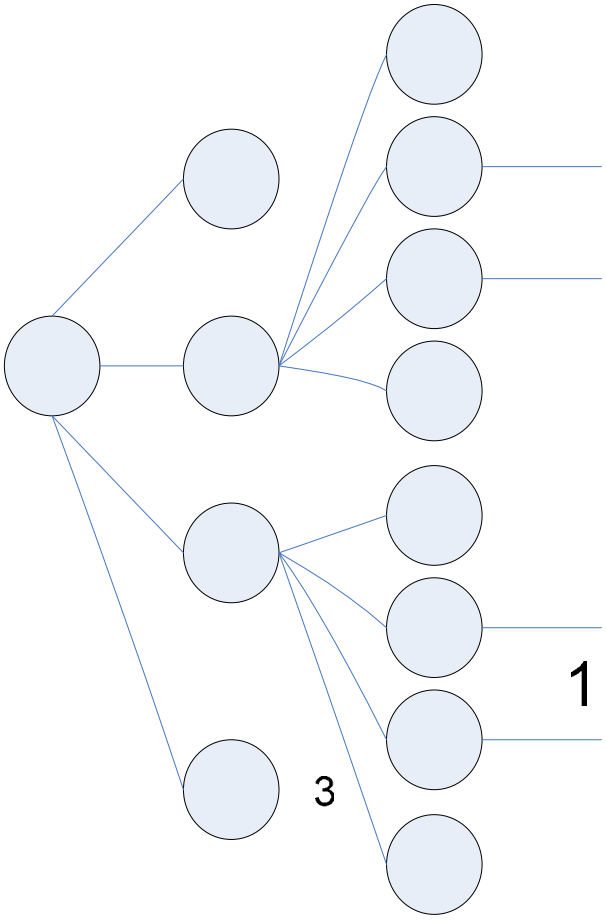
## 3. ANALISIS

Berdasarkan tahapan-tahapan diatas dapat dilihat bahwa setiap tahap memiliki kelebihan dan kekurangan. Misal tahap 1, tahap ini memiliki kelebihan yaitu pemain akan mendapatkan biji tambahan dari biji yang berada di sisi musuh, namun tahap ini hanya memungkinkan pemain hanya melakukan satu langkah.

Pada tahap 2, kelebihannya adalah pemain dapat menentukan langkah selanjutnya dengan bebas. Kekurangannya adalah pemain akan mendapatkan tambahan hanya satu demi satu kecuali pemain melanjutkan langkah "tembak".

Pada tahap 3, kelebihannya tahap ini adalah bisa melakukan kombinasi dengan tahap 1 ataupun 2, namun kekurangannya adalah jika pengecekan gagal atau tidak ada langkah yang dapat membuat pemain mendapat giliran berulang-ulang maka akan dilakukan tahap 1

ataupun 2. Lebih jelasnya akan digambarkan melalui graf di bawah ini :




**Gambar 3.a. Graf Penyelesaian Congklak**

Terlihat dari graf diatas bahwa setiap langkah hanya akan berhenti bila melakukan tahap 1 atau "menembak", atau melakukan tahap 4. Tahap 4 adalah suatu langkah yang diambil pemain, dimana jika ketiga tahapan diatas tidak memenuhi fungsi pembatas atau tidak memenuhi syarat. Tahap ke-4 merupakan tahap *default* dimana pemain hanya berusaha agar pemain tersebut tidak terlalu banyak mengisi biji di lubang-lubang sisi musuh.
Pada awalnya setiap tahap dicek terlebih dahulu mengurut secara prioritas dari tahap 1 sampai tahap 4. Lebih jelasnya bisa melihat algoritma di bawah ini :

```
While (!over)
{
    X1 = Tahap1();
    X2 = Tahap2();
    X3 = Tahap3();
    Steps = Max(x1,x2,x3);
    If(Steps == x1)
    {
        Langkah1();
        over = true;
    }
    else if (Steps == x2)
    {
        Langkah2();
    } else if (Steps == x3)
    {
        Langkah3();
    } else
    {
        Langkah4();
        over = true;
    }
```

```
//tahap1()
i = 0;
while (i < 7) && (not kosong)
{
    if(Lubang(i) == 0)
    {
        kosong = true;
    } else {
        i++;
    }
}
index = i;
while (i != 0) && (not found)
{
    i--;
    if(Lubang(i) == index - i)
    {
        found == true;
    }
}
If found
{
    return Lubang(i) + Lubang(15-i);
}
else
{
    return 0;
}
```

```
//tahap2()
I = 6;
while (i ≥ 0)
{
    if(Lubang(i) == 7 - i)
    {
        jmlhLangkah++;
    }
    else {i--;}
}
return jmlhLangkah;
```

```
//Tahap3()
Bas = 0;
For(int I = 0; I < 7; i++)
{
    While (Bas ≤ 7)
    {
        If(Lubang(Bas) < 7 || Lubang(Bas) – 15 – I <
7)
        {
            jmlhLangkah++;
        }
        If(Lubang(Bas) < 7)
        {
            Bas = Lubang(Bas) + Bas;
        } else {Bas = Lubang(Bas) – 15 + Bas;}
    }
    TabMax(i) = jmlhAngka;
}
return Max(TabMax);
```

## 4. KESIMPULAN

Kesimpulan yang dapat diambil dari makalah ini adalah :

- Untuk menentukan langkah optimal perlu meninjau beberapa langkah ke depan. Bila dalap permainan aslinya, tidak mungkin dilakukan komputasi untuk menemukan langkah optimal sampai akhir, karena langkah-langkah selanjutnya dipengaruhi oleh langkah-langkah musuh.
- Pencarian solusi ini hanya dapat dilakukan pada satu giliran hingga giliran berpindah ke musuh.
- Pencarian solusi perlu melihat dari sisi banyaknya pemain mendapatkan biji di lubang penampung dan melihat sedikitnya biji yang didapat musuh di lubang penampung.
- Relasi Rekurens
  $f_1(x_1) = max\{R_1(p_1)\}$ (*basis*)
  $f_k(x_k) = max\ \{R_k(p_k) + f_{k-1}(x_{k-1})\}$, $k = 2,3,..$ *(rekurens).*

## REFERENSI


[1] http://www.expat.or.id/info/congklakinstructions.html. Diakses tanggal 22 Mei 2007.

[2] http://www.expat.or.id/info/congklak.html. Diakses tanggal 22 Mei 2007.

[3] http://su.wikipedia.org/wiki/Congklak. Diakses tanggal 22 Mei 2007.

[2] Munir, Rinaldi., "Strategi Algoritmik", Program Studi Informatika STEI ITB, 2007.