

Penerapan Algoritma Greedy dalam Permainan Tradisional Congklak

Gisela Supardi 13515009¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganeesha 10 Bandung 40132, Indonesia

¹13515009@std.stei.itb.ac.id; giselasupardi@gmail.com

Abstrak—Indonesia merupakan negara yang kaya akan beragam bahasa dan budaya. Permainan-permainan tradisional juga merupakan kekayaan budaya yang dimiliki Indonesia. Salah satu contoh permainan tradisional yang terkenal adalah permainan congklak. Bermain congklak bukan saja memberi kesenangan namun juga dapat meningkatkan daya berpikir pemainnya. Hal ini dikarenakan untuk memenangkan permainan ini diperlukan strategi khusus agar dapat mengalahkan lawan. Dalam makalah ini akan dijabarkan mengenai pemanfaatan strategi algoritma *greedy* dalam permainan congklak. Dengan memanfaatkan algoritma *greedy*, diharapkan peluang untuk memenangkan permainan ini dapat menjadi lebih besar. Selain itu, makalah ini mungkin dapat digunakan sebagai pengembangan *artificial intelligence* untuk aplikasi permainan congklak.

Kata kunci—permainan tradisional; congklak; strategi; algoritma *greedy*.

I. PENDAHULUAN

Seiring dengan perkembangan zaman, teknologi juga semakin berkembang dengan pesat. Kebudayaan mulai ditinggalkan karena sering dianggap tidak relevan dan tidak praktis oleh masyarakat. Hal ini tidak terkecuali, berlaku pula pada permainan-permainan tradisional Indonesia. Anak-anak di Indonesia, terutama yang tinggal di daerah perkotaan, sudah asing dengan permainan-permainan tradisional Indonesia. Padahal permainan tradisional Indonesia banyak yang seru dan juga mendidik. Sebut saja sebuah permainan papan tradisional, congklak.

Permainan congklak dapat meningkatkan kemampuan berpikir pemainnya. Dalam permainan congklak, diperlukan strategi untuk dapat mengungguli lawan. Pada setiap giliran, terdapat beberapa pilihan yang memungkinkan untuk dijalankan. Pemain haruslah membuat keputusan dari pilihan-pilihan yang ada. Oleh sebab itu, algoritma *greedy* dirasa cocok untuk permasalahan ini.

Algoritma *greedy* dapat digunakan untuk mencari solusi paling optimal pada setiap kesempatan jalan. Dengan memanfaatkan algoritma *greedy*, diharapkan pemain dapat memilih pilihan yang tepat di setiap giliran jalannya sehingga kemudian dapat memenangkan permainan. Tentu saja pilihan tersebut mungkin bukan merupakan pilihan yang terbaik untuk ke depannya. Namun untuk sebuah permainan, algoritma

tersebut cocok karena selain dapat membantu memudahkan penyeleksian pilihan, algoritma *greedy* juga mudah untuk dimengerti. Karena bermain sebaiknya tidak menjadi beban pikiran tetapi sebaliknya, bermain seharusnya membantu menenangkan pikiran dan membuat senang pemainnya.

II. DASAR TEORI

A. Congklak

Permainan congklak sesungguhnya bukan merupakan permainan asli dari Indonesia. Permainan ini diadaptasi dari permainan tradisional yang berasal dari Afrika atau Arab. Oleh sebab itulah congklak memiliki berbagai macam nama. Bahkan di Indonesia sendiri, nama permainan congklak di daerah yang satu dengan daerah lainnya ada yang berbeda-beda. Di Sumatera, congklak dikenal dengan nama congkak. Sedangkan di Jawa dikenal dengan congklak atau dakon. Di Lampung, congklak dikenal sebagai dentuman lamban. Dan masih banyak lagi sebutan lain untuk congklak. Dalam Bahasa Inggris sendiri congklak disebut sebagai mancala.

Congklak merupakan permainan yang menggunakan papan. Bahkan beberapa ahli percaya bahwa congklak merupakan permainan papan tertua di dunia. Papan pada permainan congklak terdiri dari tujuh lubang kecil pada masing-masing sisi dan satu lubang yang lebih besar pada ujung-ujung papan yang biasa disebut lumbung. Sehingga bila dijumlahkan terdapat empat belas lubang kecil dan dua lubang besar pada papan congklak. Papan congklak yang dikenal luas saat ini umumnya terbuat dari kayu. Ada yang diukir pada ujung-ujungnya mau pun papan polos biasa.

Untuk memainkan congklak, dibutuhkan alat lain yang sering disebut biji congklak atau buah congklak. Ada pun biji congklak di Indonesia umumnya merupakan cangkang-cangkang kerang kecil seperti pada gambar 1. Namun jika tidak ada cangkang kerang, biji-bijian atau pun batu-batu kecil juga umum digunakan sebagai biji congklak.

Permainan congklak umumnya dikenal sebagai permainan perempuan di Indonesia. Oleh sebab itu jarang ada laki-laki yang memainkan permainan ini. Congklak sendiri merupakan permainan untuk 2 orang. Masing-masing orang berada di salah satu sisi papan. Semua lubang kecil di sisinya dan lumbung di kiri dianggap milik pemain tersebut. Pada awalnya kedua pemain akan bermain bersamaan sampai salah satu pemain mati.

Kemudian permainan akan digilir. Pemain yang saat itu mati harus menunggu hingga lawannya mati untuk mendapat giliran main.

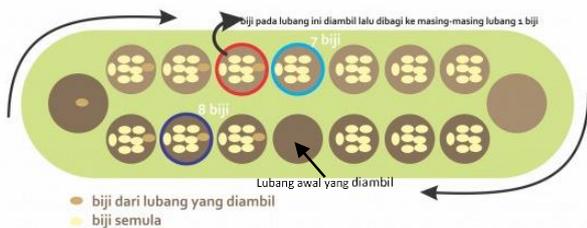


Gambar 1 Contoh papan dan biji congklak

Sumber:

<http://juanmarliyahadi.blogspot.co.id/2015/10/permainan-congklak.html>

Tujuan dari permainan congklak adalah mengumpulkan biji congklak sebanyak-banyaknya pada lumbung. Pada awalnya lumbung kosong dan lubang kecil masing-masing diisi biji congklak sebanyak jumlah lubang kecil per sisi. Cara bermainnya adalah memilih untuk menjalankan biji dari sisi sendiri. Pemain menjalankan biji congklak searah jarum jam. Setiap lubang yang dilewati, kecuali lumbung lawan, harus diisi dengan sebuah biji congklak dari yang dipegang. Jika biji terakhir jatuh di lumbung, kita dapat memilih lagi dari lubang kita. Jika biji terakhir jatuh di lubang berisi biji lainnya, maka semua isi lubang tersebut diambil dan dijalankan lagi. Sedangkan jika jatuh pada lubang kosong, giliran kita habis dan lawan berhak mendapat giliran jalan. Apabila kita mati di sisi sendiri dan pada lubang lawan yang bersebrangan terdapat biji congklak, kita berhak mengambil seluruh biji pada lubang lawan itu beserta dengan satu biji yang ada di sisi kita.



Gambar 2 Visualisasi cara bermain congklak

Sumber: <http://blog.unnes.ac.id/ituini/congklak-atau-dakon/dengan-ubahan-seperlunya>

Apabila saat meletakkan biji terakhir pada salah satu sisi tidak ada lagi biji congklak yang tersisa, maka permainan tersebut akan dihentikan. Pemain tanpa biji congklak akan dinyatakan "kalah jalan". Permainan dapat dilanjutkan dengan mengisi kembali lubang pada sisi sendiri dari lumbung sendiri. Jika kemudian tidak cukup diisi sejumlah yang seharusnya, lubang akan dibiarkan kosong dan tidak perlu diisi selama permainan berikutnya berlangsung. Untuk ronde 2 dan seterusnya, pemain yang sebelumnya kalah jalan tidak boleh bermain di awal. Ia harus menunggu lawannya yang sebelumnya menang jalan mati terlebih dahulu. Pemain dinyatakan menang

mutlak apabila pada permainan selanjutnya lawannya tidak dapat mengisi satu pun lubang di sisinya.

B. Algoritma Greedy

Algoritma greedy merupakan algoritma yang menyelesaikan suatu permasalahan secara langkah per langkah. Pada setiap langkahnya, terdapat beberapa opsi yang dapat dipilih. Algoritma greedy sendiri berprinsip memilih pilihan yang terbaik pada tiap langkah (optimum lokal) dengan harapan akan berakhir dengan solusi optimum global. Prinsip ini dinamakan "take what you can get now". Namun hal ini tidak berarti bahwa solusi akhir yang tercipta dari algoritma greedy merupakan solusi terbaik. Pada kenyataannya, algoritma ini tidak beroperasi secara menyeluruh terhadap semua alternatif solusi sehingga belum tentu solusi yang dipilih merupakan solusi paling optimal.

Walau pun solusi dari algoritma ini memiliki kekurangan, namun algoritma ini jauh lebih efektif dibandingkan *exhaustive search*, atau yang lebih dikenal dengan nama *brute force*, sehingga tetap sering digunakan untuk menyelesaikan persoalan. Terutama untuk masalah-masalah optimasi, yakni persoalan maksimasi dan minimasi.

Ada lima elemen dalam algoritma greedy, yaitu sebagai berikut.

1. Himpunan kandidat (C)

Himpunan kandidat adalah opsi-opsi yang dapat dipilih pada setiap pilihan.

2. Himpunan solusi (S)

Himpunan solusi merupakan kumpulan solusi yang dihasilkan pada tiap langkah yang diharapkan akan membentuk solusi optimal. Himpunan solusi berisi elemen-elemen dari himpunan kandidat yang terpilih.

3. Fungsi seleksi

Fungsi seleksi merupakan fungsi yang menghasilkan solusi per langkah yang diharapkan dapat membentuk solusi optimal. Dengan kata lain cara untuk melakukan pemilihan opsi sehingga menjadi solusi.

4. Fungsi kelayakan (*feasible*)

Fungsi kelayakan adalah batasan-batasan dalam membentuk solusi dan menjadi penentu apakah opsi tersebut dapat dipilih. Fungsi kelayakan dapat dikatakan sebagai tolok ukur apakah pilihan pada langkah tersebut melanggar *constraint* yang ada.

5. Fungsi objektif

Fungsi objektif fungsi yang bertujuan untuk memaksimalkan atau meminimumkan solusi yang dipilih.

Contoh penerapan algoritma greedy yang paling sederhana adalah persoalan penukaran uang. Misalkan ada koin dengan nilai 25, 10, 5, dan 1. Diminta untuk menukar uang senilai 32 dengan jumlah koin sesedikit mungkin. Maka, himpunan kandidat adalah koin bernilai 25, 10, 5, dan 1. Himpunan solusi adalah koin-koin yang nantinya akan dipilih dan nilainya sama dengan 32. Fungsi seleksinya adalah memilih koin bernilai tertinggi dari himpunan kandidat yang tersisa. Fungsi

kelayakannya adalah nilai dari koin yang dihasilkan haruslah kurang dari sama dengan 32. Sedangkan fungsi objektifnya adalah jumlah koin yang minimum. Dengan algoritma greedy, persoalan tersebut dapat diselesaikan dengan cara memilih koin bernilai paling besar terlebih dahulu. Kemudian apabila jumlah nilai tidak lolos dari fungsi kelayakan maka akan dicoba koin bernilai terbesar selanjutnya. Maka, akan didapatkan hasil 4 koin dengan himpunan solusi $\{25, 5, 1, 1\}$.

Dalam dunia pemrograman, sebelum melakukan algoritma greedy akan lebih baik jika dilakukan pengurutan opsi tertentu terlebih dahulu. Dengan dilakukannya pengurutan, pemilihan opsi akan menjadi lebih mudah.

III. PEMBAHASAN

A. Elemen Algoritma Greedy dalam Congklak

Sebelum mengimplementasikan algoritma greedy ini, akan lebih baik jika terlebih dahulu didefinisikan elemen-elemennya pada persoalan yang sedang dikerjakan. Oleh sebab itu, akan dicari elemen algoritma greedy dalam permainan congklak.

1. Himpunan kandidat

Himpunan kandidat adalah lubang-lubang kecil yang berada di sisi papan. Namun karena lubang yang dimainkan adalah lubang yang berisi biji congklak, maka berarti himpunan kandidatnya adalah lubang kecil yang berisi biji congklak.

2. Himpunan solusi

Lubang-lubang yang dipilih untuk dijalankan. Karena langkah yang optimal selalu bergantung pada gerakan lawan, dan sebaliknya gerakan lawan bergantung pada pilihan kita, maka tidak ada himpunan solusi yang pasti untuk setiap permainannya.

3. Fungsi seleksi

Fungsi ini adalah fungsi yang digunakan untuk menentukan lubang mana yang akan dipilih untuk mencapai solusi optimum. Pada permainan congklak, pemilihan ini didasari oleh banyak faktor. Karena itu untuk fungsi ini akan dijelaskan secara terpisah pada bagian IIB.

4. Fungsi kelayakan

Lubang yang dipilih harus berada pada sisi pemain, bukan sisi lawan. Lubang yang dipilih untuk dimainkan hanya boleh satu setiap kali jalan (saat mendapat giliran atau pun saat biji terakhir jatuh di lumbung).

5. Fungsi objektif

Langkah yang diambil jumlahnya minimum.

B. Fungsi Seleksi pada Congklak

Untuk memenangkan permainan, yang terpenting adalah mengisi lumbung sebanyak-banyaknya. Hal ini dapat dicapai dengan cara mempertahankan diri agar tetap hidup selama mungkin. Dengan semakin lama hidup maka semakin banyak putaran yang dicapai dan akhirnya lumbung pun akan semakin banyak terisi. Jika memungkinkan, jalan yang terbaik adalah selalu memilih jalan yang langsung masuk ke lumbung. Dengan memilih jalan tersebut maka kita tidak perlu membagikan biji

kita ke daerah lawan. Tujuannya adalah untuk mencegah kita kalah jalan.

Apabila tidak ada jalan yang memungkinkan, dengan kata lain semua pilihan akan menyebabkan mati dengan satu kali mengisi lumbung atau tidak mengisi lumbung sama sekali, maka lebih baik memilih untuk mati namun biji terakhir jatuh di tempat sendiri sehingga bisa mengambil biji dari lubang lawan. Langkah ini umumnya disebut "nembak". Saat nembak pun, perlu dipilih lubang yang seberangnya memiliki biji terbanyak agar biji yang dapat diambil semakin banyak.

Kadang kala ada juga saat di mana pilihan yang terbaik justru nembak dibandingkan harus jalan. Hal ini umumnya terjadi apabila lawan kita mengumpulkan banyak sekali biji dalam salah satu lubang yang dimilikinya sedangkan pada lubang lainnya banyak yang kosong sehingga lebih rentan mati.

Saat permainan akan berakhir, umumnya hanya tersisa sedikit sekali biji pada lubang-lubang kecil. Di saat seperti ini prioritas berubah menjadi bertahan hidup paling lama agar tidak sampai kalah jalan. Menang jalan akan membantu mempermudah kemenangan di ronde berikutnya walau di ronde saat ini kalah sekali pun.

Meninjau dari kondisi-kondisi tersebut, dapat dibuat beberapa fungsi seleksi untuk masing-masing kondisi tertentu.

1. Kondisi saat banyak lubang yang terisi

Kondisi ini biasanya hanya terjadi pada saat awal-awal permainan. Pada kondisi ini, yang perlu dilakukan adalah memilih lubang yang dapat mengantarkan tepat masuk ke lumbung. Apabila tidak ada yang membuat tepat masuk, pilihlah lubang yang dapat mengantarkan kita untuk sampai ke daerah lawan agar dapat mengambil dan membagikan biji yang dimiliki lawan ke daerah kita sendiri. Tetapi yang mana pun lubang yang dipilih, jangan sampai membuat kita mati.

2. Kondisi saat sedikit lubang yang terisi dan lawan menimbun biji pada sedikitnya 1 lubang

Kondisi ini umumnya tercapai di pertengahan ronde. Saat pertengahan biasanya secara tidak sadar akan ada lubang yang tidak tersentuh sehingga semua biji tertumpuk dalam lubang tersebut. Jika hal tersebut terjadi, usahakan membuat lubang di sisi kita menjadi kosong untuk nembak lubang lawan.

3. Kondisi saat sedikit lubang yang terisi baik di lawan mau pun di tempat pemain

Di saat seperti ini, penting untuk menjaga agar lubang di tempat kita tetap terisi. Jadi, lubang yang harus dipilih adalah yang mengantar kita untuk menyebar biji yang tersisa ke lubang-lubang lainnya.

4. Kondisi mendekati akhir permainan

Kondisi yang dimaksud adalah ketika jumlah keseluruhan biji yang tersisa bahkan kurang dari jumlah seluruh lubang kecil yang ada. Pada saat seperti ini, perlu dilakukan cara-cara untuk bertahan hidup yaitu sedapat mungkin masukkan biji-biji yang ada ke lumbung. Kemudian jika tidak ada lagi, jalan perlahan dari lubang terkanan. Perlahan yang dimaksud di sini adalah memilih untuk mati pada setiap giliran.

Usahakan agar biji di tempat lawan habis terlebih dahulu dibandingkan biji yang kita miliki. Semakin banyak kita memberi kesempatan lawan untuk berjalan maka akan semakin besar kemungkinannya kita terhindar dari kalah jalan.

5. Kondisi-kondisi lainnya yang tidak dicantumkan di atas

Apabila ada kondisi-kondisi khusus yang mungkin terjadi, selalu usahakan untuk mengisi lubang dan jangan sampai mati di tempat lawan karena kita tidak akan mendapatkan apa pun.

C. Implementasi Algoritma Greedy pada Permainan Congklak

Untuk memudahkan pembahasan, permainan congklak ini akan diubah ke dalam bahasa pemrograman yaitu dengan mendefinisikan variabel, fungsi, dan prosedur. Pertama akan didefinisikan terlebih dahulu variabel-variabel yang akan digunakan selama permainan.

1. lubangPemain

Variabel ini dimaksudkan untuk menyimpan kondisi dari lubang yang berada di sisi pemain. Array ini menyimpan integer yang merepresentasikan jumlah biji yang ada di setiap lubangnya.

2. lubangLawan

Sama dengan lubangPemain, variable ini juga menyimpan jumlah biji yang ada, namun dari sisi lawan.

3. lubangPemain

Variabel ini digunakan untuk mengetahui ada berapa banyak biji di dalam lubang pemain.

4. lubangLawan

Variabel ini digunakan untuk mengetahui ada berapa banyak biji di dalam lubang pemain.

5. jumlahJalan

Variabel ini digunakan untuk mengetahui jumlah biji yang berada di tangan pemain yang akan diedarkan.

6. posisiJalan

Variabel ini digunakan untuk mengetahui posisi pemain saat sedang menjalankan biji. Direpresentasikan sebagai integer. Range dari negatif jumlah lubang pada satu sisi hingga positif jumlah lubang satu sisi. Angka negatif merepresentasikan posisi berada di tempat lawan, angka 0 berarti di lubang, dan angka positif berarti berada di area sendiri. Untuk angka negatif, semakin negatif berarti semakin dekat dengan lubang lawan. Sedangkan untuk angka positif maka semakin positif berarti semakin jauh dari lubang sendiri.

Kemudian, berikut adalah predikat-predikat, yaitu fungsi yang menghasilkan *boolean*, beserta penjelasannya untuk penggunaan dalam pseudocode.

1. endGame

Predikat untuk menentukan apakah ronde berakhir.

2. isGiliran

Predikat ini menyatakan apakah pemain sedang mendapatkan giliran jalan. Bernilai true jika pemain masih hidup dan bernilai false jika jatuh ke lubang kosong.

3. isIsi

Predikat ini menyatakan kondisi dari sebuah lubang. Bernilai true jika lubang yang sedang diperiksa memiliki biji di dalamnya dan false jika lubang tersebut kosong.

4. isSeberangIsi

Predikat ini digunakan untuk mengecek kondisi dari lubang lawan. Namun input indeks akan menghasilkan indeks lawan yang sebaliknya yang berada di seberang satu langkah setelah indeks yang dimasukkan. Misalkan n adalah jumlah lubang di satu sisi dan i adalah indeks input, maka yang dicek adalah lubangLawan dengan indeks n-i. Prekondisi untuk predikat ini adalah i lebih besar sama dengan 1.

5. isKondisiSatu

Predikat ini untuk mengecek apakah keadaan permainan saat itu memenuhi kondisi yang diberikan pada subbab IIC poin 1.

6. isKondisiDua

Predikat ini untuk mengecek apakah keadaan permainan saat itu memenuhi kondisi yang diberikan pada subbab IIC poin 2.

7. isKondisiTiga

Predikat ini untuk mengecek apakah keadaan permainan saat itu memenuhi kondisi yang diberikan pada subbab IIC poin 3.

8. isKondisiEmpat

Predikat ini untuk mengecek apakah keadaan permainan saat itu memenuhi kondisi yang diberikan pada subbab IIC poin 4.

9. isMasukLubung

Predikat ini untuk mengecek apakah jika memilih lubang x dapat tepat masuk ke lubang.

Kemudian, berikut adalah pseudocode yang dapat diterapkan pada permainan congklak.

Kamus

```
// deklarasi variabel global serta predikat yang telah disampaikan di luar pseudocode ini
// prosedur-prosedur lain
procedure jalan (input posJalan: integer, output hidup: boolean)
{prosedur yang merupakan pengedaran biji}
  if posJalan > 0
    jmlJalan = lubangPemain[posJalan - 1]
  else jmlJalan = lubangLawan[n - posJalan]
  endif
  while jmlJalan > 0 do
    if posJalan > 0
```

```

        lubangPemain[posJalan - 1] += 1
        posJalan -= 1
    else if posJalan = 0
        lumbungPemain += 1
        posJalan -= 1
    else // posJalan < 0
        lubangLawan[n + posJalan] += 1
        if posJalan = -n
            posJalan = n
        else
            posJalan -= 1
        jmlJalan -= 1
    endwhile
    if posJalan = n
        posJalan = -n-1
    hidup = isIsi(posJalan+1)
    // biji di tangan sudah habis; jmlJalan = 0

function prediksiBijiTerakhir (jml: integer,
pos: integer) → integer
{fungsi untuk menentukan di lubang mana biji
terakhir akan jatuh}
while jml > 0 do
    if pos > 0
        pos -= 1
    else if pos = 0
        pos -= 1
    else // pos < 0
        if pos = -n
            pos = n
        else
            pos -= 1
        jml -= 1
    endwhile
    if pos = n return -n
else return pos+1

procedure greedy (output hidup: boolean)
{prosedur untuk memilih lubang yang ingin
dijalankan dengan menerapkan algoritma greedy}
arrKemungkinan: array of integer
for i = 0 to n do
    arrKemungkinan[i] = prediksiBijiTerakhir
    (lubangPemain, i)
endfor
if isKondisiSatu
    for i = 0 to n do
        if isMasukLumbung(i)
            jalan(i, hidup)
        endif
    endfor
    // tidak ada yang dapat langsung masuk ke
    lumbung
    i = indeks dengan nilai paling negatif
    dari arrKemungkinan
    if isIsi(i) jalan(i, hidup)
    else i = indeks dengan nilai ternegatif
    selanjutnya
    repeat until mendapat i
else if isKondisiDua
    i = cariLubangTimbun
    kosongkanLubang(-i)
    jalan(cariJalanKeLubang(-i), hidup)
else if isKondisiTiga
    i = indeks dengan nilai paling banyak dari
    lubangPemain
    if
    isIsi(prediksiBijiTerakhir(lubangPemain
    (i), i)) jalan(i, hidup)
    else i = indeks dengan nilai terbanyak
    selanjutnya
    repeat until mendapat i

```

```

else if isKondisiEmpat
    for i = 0 to n do
        if isMasukLumbung(i)
            jalan(i, hidup)
        endif
    endfor
    i = indeks lubang yang membuat mati
else
    for i = 0 to n do
        if isMasukLumbung(i)
            jalan(i, hidup)
        endif
    endfor
    i = indeks dengan nilai paling banyak dari
    lubangPemain
    if
    isIsi(prediksiBijiTerakhir(lubangPemain
    (i), i)) jalan(i, hidup)
    else i = indeks dengan nilai terbanyak
    selanjutnya
    repeat until mendapat i

```

Algoritma

```

inisialisasiVariabelGlobal
endGame = true
isGiliran = true
while not endGame
    while isGiliran
        greedy(hidup)
        if not hidup isGiliran = false
        endif
    endwhile
    waitForLawan
    endGame = salah satu sisi tidak memiliki
    biji lagi
endwhile

```

IV. PENGEMBANGAN PENERAPAN ALGORITMA GREEDY PADA PERMAINAN CONGKLAK

Seperti yang kita ketahui bersama, dewasa ini masyarakat pada umumnya cenderung lebih memilih untuk memainkan aplikasi-aplikasi permainan pada gawai. Hal tersebut karena permainan tradisional dinilai tidak praktis. Apalagi permainan congklak yang membutuhkan papan yang cukup besar serta sedikitnya 98 biji congklak untuk dapat dimainkan. Masalah ini jika dibiarkan dapat mengikis kebudayaan Indonesia dan membuat generasi-generasi setelahnya tidak mengenal permainan congklak ini.

Untuk mengatasinya perlu dilakukan inovasi yaitu menggabungkan elemen tradisional dengan elemen modern. Dengan kata lain, permainan congklak ini perlu dibuat aplikasinya agar tidak hilang ditelan zaman.

Dalam membuat aplikasi permainan congklak ini, alangkah baiknya dibuat dalam 2 mode, yaitu *player versus player* dan juga *player versus computer*. Algoritma greedy pada congklak ini dapat diimplementasikan untuk membuat *artificial intelligence* sebagai lawan dalam aplikasi permainan congklak ini. Keuntungan saat menerapkan algoritma greedy dalam permainan ini adalah kemampuan untuk menghitung jumlah biji congklak dan memprediksi langkah untuk semua pilihan yang tidak dapat dilakukan pada saat bermain di *real life*.

Untuk aplikasi permainan ini sendiri saya telah menemukan beberapa contohnya baik yang online mau pun yang offline. Dapat ditemukan dengan mudah pada mesin pencari dengan mengetikkan “game congklak” atau “game Mancala”.



Gambar 3 Contoh aplikasi permainan congklak online

V. SIMPULAN

Berdasarkan pemaparan dalam makalah ini, dapat ditarik simpulan bahwa algoritma greedy merupakan algoritma yang cukup mangkus untuk membantu dalam permainan congklak. Hal ini dikarenakan greedy mencari solusi langkah per langkah sesuai dengan permainan congklak yang dinamis serta tidak rumit untuk diterapkan. Selain itu, algoritma greedy dalam permainan congklak juga dapat dikembangkan menjadi dasar pembuatan *artificial intelligence* dalam permainan congklak pada gawai. Diharapkan jika diadaptasi menjadi aplikasi gawai, ketertarikan masyarakat terhadap permainan congklak dapat kembali meningkat.

ACKNOWLEDGMENT

Pertama-tama saya ingin mengucapkan terima kasih kepada Tuhan Yang Maha Esa yang hanya karena berkat

penyertaannya saya dapat menyelesaikan makalah ini. Kemudian tidak lupa saya mengucapkan terima kasih kepada dosen-dosen pengampu mata kuliah IF2211 Strategi Algoritma, yaitu Bapak Dr. Ir. Rinaldi Munir, MT., Ibu Masayu Leylia Khodra, S.T, M.T, serta Ibu Dr. Nur Ulfa Maulidevi, S.T., M.Sc. yang telah membimbing saya dalam belajar selama satu semester ini dan memungkinkan saya untuk membuat makalah ini. Selanjutnya ucapan terima kasih juga ingin saya sampaikan kepada kedua orang tua saya yang telah mendukung perkuliahan saya hingga di penghujung semester 4 ini. Terakhir, saya juga ingin mengucapkan terima kasih kepada teman-teman Teknik Informatika 2015, khususnya kepada Ferdinandus Richard, yang telah memberi dukungan baik secara moral maupun sosial sehingga makalah ini dapat saya selesaikan dan kumpulkan tepat pada waktunya.

REFERENCES

- [1] <http://www.expat.or.id/info/congklak.html>. Diakses pada tanggal 18 Mei 2017.
- [2] Munir, Rinaldi. 2009. Diktat Kuliah IF2211 Strategi Algoritma. Bandung: Institut Teknologi Bandung.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Jakarta, 18 Mei 2017

Gisela Supardi 13515009