

PENERAPAN ALGORITMA A* UNTUK PENCARIAN SOLUSI TERBAIK PADA GAME CONGKLAK

Ahmad Soleh Siregar¹, Surya Darma Nasution²

¹ Mahasiswa Teknik Informatika STMIK Budi Darma

² Dosen Tetap STMIK Budi Darma

^{1,2} Jl. Sisingamangaraja No. 338 Simpang Limun, Medan

ABSTRAK

Congklak adalah sebuah permainan tradisional Indonesia dengan menggunakan media kayu, batu, tanah dan juga benda lain yang dapat membentuk 16 lobang melingkar, yang dapat dimainkan hanya dua orang saja. Dalam permainan ini pemain memasukkan buah congklak kedalam lobang masing-masing satu dan akan berhenti jika buah congklak habis dalam lobang yang kosong. Namun dengan perkembangan jaman permainan ini bisa saja hilang disebabkan terlalu banyaknya permainan dibuat dalam bentuk digital, hanya dalam sebuah aplikasi permainan dapat dimainkan. Agar permainan ini tidak hilang penulis berusaha membangun permainan congklak ini dalam bentuk digital, dan menerapkan suatu metode untuk dapat mencari jalan terbaik dalam permainan congklak tersebut. Metode yang digunakan adalah metode A*, dimana metode ini bekerja untuk mencari jalan terbaik dalam suatu pencarian masalah dalam permainan, dan bahasa pemrograman yang digunakan adalah Borland Delphi, dimana program ini dapat membuat aplikasi permainan congklak dan menerapkan metode A* dengan mudah. Dengan aplikasi Borland Delphi permainan congklak dapat dimainkan dalam bentuk digital dan mudah dibawa kemana saja tanpa harus menggunakan media kayu sebagai papan congklak, permainan congklak dalam aplikasi ini juga dapat bermain sendiri dan bermain melawan sistem yang bergerak mengikuti instruksi-instruksi dalam aplikasi. Dengan penerapan metode A* dalam permainan ini juga menunjukkan jalan terbaik dalam permainan congklak.

Kata Kunci: Congklak, metode A*, Borland Delphi

I. PENDAHULUAN

Game atau permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius dengan tujuan refresing. Bermain game sudah dapat dikatakan sebagai life style masyarakat dimasa kini (Nofarianto Sihite, 2016).

Congklak atau biasa disebut dakon merupakan permainan tradisional yang hampir dikenal diseluruh Indonesia walaupun memiliki nama yang berbeda di setiap daerah tetapi cara bermainnya tetap sama, permainan congklak biasanya dimainkan minimal 2 (dua) orang pemain yang bermain bergantian jika biji congklaknya tidak lebih dari 1, terdapat suatu permasalahan jika dilihat pada permainan ini yaitu tidak optimalnya cara bermain jika dimainkan oleh user (manusia) karena user biasanya memilih biji congklak yang banyak secara logika, berbeda dengan komputer jika diterapkan suatu algoritma maka komputer akan memilih solusi optimal tidak berdasarkan jumlah biji congklak tetapi lebih kepada proses penyelesaian dengan tepat, untuk itu penulis mencoba menyelesaikan permasalahan permainan congklak dengan menggunakan algoritma A*

Banyak metoda AI atau kecerdasan buatan yang dapat diterapkan untuk menyelesaikan problema tersebut. Salah satunya adalah dengan menggunakan bantuan pohon pelacakan. Kondisi-kondisi (*state-state*) yang mungkin digambarkan dalam suatu pohon pencarian dimulai dari mendefinisikan kondisi awal (*start state*) sebagai

akar dari pohon biner. Proses dilanjutkan dengan menggambarkan kondisi (*state*) berikutnya dari *state* tersebut hingga didapatkan kondisi yang diinginkan (*goal state*), karena itu Kecerdasan buatan atau *artificial intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia (Ardhan Wahyu, Kecerdasan Buatan Untuk Menyelesaikan Rubik's Cube Dengan Algoritma IDA, Universitas Negeri Malang).

A* mendeskripsikan proses pemecahan masalah dimana seseorang harus mengambil setiap keputusan terbaik pada setiap langkahnya. Optimisasi dengan dynamic programming yang diterapkan di A* dicapai dengan memilih setiap keputusan terbaik (optimal) dalam setiap tahap. Dalam penyelesaian Menara Hanoi terdapat beberapa algoritma yang bisa digunakan, termasuk salah satunya adalah Pemrograman Dinamis (*Dynamic Programming*).

Langkah yang digunakan untuk menyelesaikan dengan Berdasarkan uraian di atas, penulis ingin merancang suatu perangkat lunak game yang mampu untuk melakukan penyelesaian masalah permainan congklak.

II. LANDASAN TEORI

A. Algoritma A* (A-Star)

Pencarian buta (*uninformed* atau *blind search*) umumnya kurang efisien. Hal ini disebabkan oleh waktu akses yang cukup lama dan besarnya memori yang diperlukan. Untuk mengatasi

kelemahan ini maka penambahan domain pengetahuan akan menghasilkan suatu proses pencarian dan investigasi baru. Pencarian tersebut biasanya diistilahkan dengan *informed search*.

George Poyla mendefinisikan *AStar* sebagai studi metode dan kaidah penemuan. Dalam pencarian ruang keadaan, *AStar* dinyatakan sebagai aturan untuk melakukan pemilihan cabang-cabang dalam ruang keadaan yang paling tepat untuk mencapai solusi permasalahan yang dapat diterima.

Pemecahan masalah dalam AI menggunakan heuristik dalam dua situasi dasar, yaitu:

1. Permasalahan yang mungkin tidak mempunyai solusi yang pasti disebabkan oleh ambiguitas (keraguan / ketidakpastian) mendasar dalam pernyataan permasalahan atau data yang tersedia. Diagnosa kedokteran merupakan salah satu contohnya, dimana sejumlah gejala mungkin dapat ditimbulkan oleh berbagai macam penyebab yang mungkin. Dokter menggunakan heuristik *AStar* untuk memilih atau menentukan diagnosa yang paling dapat diharapkan dan memutuskan rencana penanganannya. *Vision* merupakan masalah lainnya untuk permasalahan ketidakpastian yang mendasar.
2. Permasalahan yang boleh jadi memiliki solusi pasti, tetapi biaya komputasinya untuk mendapatkan solusi tersebut mungkin sangat tinggi. Dalam banyak masalah seperti catur, ruang keadaan bertambah secara luar biasa seiring dengan pertambahan jumlah keadaan yang dimungkinkan.

Sayangnya, seperti semua kaidah penemuan lainnya, heuristik *AStar* juga dapat salah. Heuristik *AStar* hanyalah panduan informasi untuk menebak langkah berikutnya yang harus diambil dalam menyelesaikan suatu permasalahan, dan sering dilakukan berdasarkan eksperimen/percobaan atau secara intuisi. Oleh karena, menggunakan informasi yang terbatas, heuristik *AStar* jarang dapat memprediksi tingkah laku yang eksak dari ruang keadaan saat dilakukan pencarian. Heuristik *AStar* dapat membimbing algoritma pencarian untuk mendapatkan solusi suboptimal atau gagal menemukan solusi apapun.

Heuristik *AStar* dan perancangan algoritma untuk mengimplementasikan pencarian heuristik telah menjadi inti permasalahan penelitian I. *Game playing* dan pemecahan teorema (*teorem solving*) adalah dua aplikasi paling tua dari AI, kedua-duanya memerlukan heuristik untuk memangkas ruang dari solusi yang mungkin.

Saat ini, penelitian mengenai sistem pakar telah menyadari arti penting heuristik *astar* sebagai komponen yang esensial dalam pemecahan permasalahan. Bila seorang pakar memecahkan suatu permasalahan maka dia akan menguji informasi yang tersedia dan membuat suatu keputusan (Yulikus Partono, 2008).

A. Congklak

Congklak atau Dakon adalah suatu permainan tradisional, di luar Jawa permainan ini lebih dikenal dengan nama congklak. Secara internasional permainan congklak disebut Mancala berasal dari bahasa Arab 'Naqala' yang artinya bergerak. Menurut dugaan asalnya dari Jordania setelah ditemukan papan mainan yang berusia 7000-5000 tahun sebelum masehi. Sedang di Afrika permainan ini juga dikenal dengan nama Oware. Permainan dakon sampai ke Indonesia pertama kali dibawa oleh pendatang dari Arab yang rata-rata datang ke Indonesia untuk berdagang dan dakwah (Devinta Ristanti, 2011).

Permainan congklak dilakukan oleh dua orang. Ada 12 lubang dalam congklak, 2 lubang besar disebut lumbung yang merupakan penyimpanan biji yang dibagi secara merata dalam urutan arah (kanan). Jumlah biji yang dimainkan masing-masing ada 5 biji disetiap lubang. Aturan dasar misalnya lumbung mati, artinya kalau biji sampai di lumbung mati (tidak jalan lagi). Kondisi mati juga terjadi jika biji jatuh di lubang tanpa ada biji yang lain. Kondisi tersebut dapat menguntungkan player musuh karena otomatis player musuh akan langsung memulai permainannya. Pada masa lalu, permainan ini sangat lazim dimainkan oleh anak-anak bahkan remaja wanita. Tidak ada yang tahu mengapa permainan ini identik dengan dunia wanita. Menurut beberapa pendapat, permainan ini identik atau berhubungan erat dengan wanita karena permainan ini berhubungan dengan manajemen atau pengelolaan keuangan. Pada masa lalu (bahkan hingga kini) Kaum hawa disadari atau tidak, berperan penting dalam pengelolaan keuangan rumah tangga. Congklak dianggap menjadi sarana pelatihan terhadap pengelolaan atau manajemen keuangan tersebut (Devinta Ristanti, 2011).

III. ANALISIS DAN PEMBAHASAN

Congklak merupakan permainan yang membutuhkan strategi agar permainan cepat selesai dan pemain menang dengan cepat, proses penyelesaian ini membutuhkan algoritma untuk menyelesaikan masalah, untuk itu penulis menerapkan algoritma A^* sebagai algoritma yang digunakan.

Prosedur pencarian A^* merupakan pencarian yang dilakukan dengan mengunjungi tiap-tiap *node* secara sistematis pada setiap level hingga keadaan tujuan (*goal state*) ditemukan. Atau dengan kata lain, penelusuran dilakukan dengan mengunjungi *node-node* per level hingga ditemukan *goal state*. Apabila terdapat solusi, pencarian A^* menjamin ditemukannya solusi dengan lintasan terpendek (*shortest path*). Penerapan algoritma A^* pada permainan congklak dapat dilihat pada contoh berikut. $C1=C$, $C2=A$, $C3=B$

1. State awal I:
 1. Lambung-1: 7
 2. Lambung -2: 7
 3. Lambung -3: 7
2. State awal II:
 1. Lambung -1: 5, 2 Buah biji congklak sudah ada di lambung 2 dan lambung 3
 2. Lambung -2: 8
 3. Lambung -3: 8
3. State awal III:
 1. Lambung -1: 5. 2 Buah biji congklak sudah ada di lambung 2 dan lambung 3
 2. Lambung -2: 7. 1 Buah biji congklak sudah ada di lambung 3
 3. Lambung -3: 9
4. State awal IV:
 1. Lambung -1: 6, 2 Buah biji congklak sudah ada di lambung 2 dan lambung 3 dan 1 biji lambung 3 berada dibagian ini
 2. Lambung -2: 8, 1 biji lambung 3 berada dibagian ini
 3. Lambung -3: 7

Game congklak akan berupa ordo $X \times Y$, supaya pengguna bisa menentukan sendiri orde yang diinginkan. Pada ordo $X \times Y$, 2 kotak akan dipakai oleh titik awal dan titik tujuan, sisanya untuk menghasilkan jalur agar terlihat jalurnya yang pada akhirnya akan menentukan jalur terpendek ke titik tujuan. Maksimal penghalang pada tiap ordo dapat kita tentukan yaitu dengan rumus pada tabel 1.

Tabel 1. Maksimal Penghalang

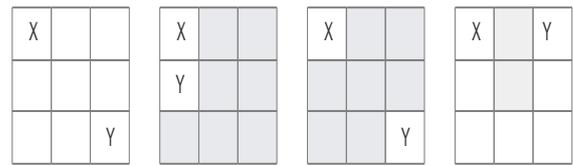
Ordo	Maksimal Penghalang
$X \times Y$	$(M \times N) - 2$
3 x 3	$(3 \times 3) - 2 = 7$
4 x 4	$(4 \times 4) - 2 = 14$
5 x 5	$(5 \times 5) - 2 = 23$
... x ...	$(... \times ...) - 2 = ...$

Keterangan :

X : Nilai baris

Y : Nilai kolom

Dengan maksimal penghalang $(X \times Y) - 2$, maka akan menghasilkan jalur yang tidak terlihat disebabkan jarak antara titik awal dan titik tujuan saling berdekatan, tapi dengan kondisi titik awal dan titik tujuan saling berjauhan maka akan menghasilkan jalur tidak akan ditemukan. Agar terlihat jalur yang dihasilkan maka diusahakan titik awal dengan titik tujuan tidak saling berdekatan dan jumlah penghalang yang akan dipasang kurang dari maksimal, implementasinya dapat dilihat dengan contoh ordo 3×3 pada gambar 3.3.



Gambar 1. Ruang peta ordo 3×3 dengan tiga kondisi

Maksimal penghalang yang akan dipasang pada ordo lainnya adalah sebanyak ordo tersebut dikurangi dua untuk menempatkan titik awal dan titik tujuan. Pada gambar 3.3 menunjukkan suatu ruang (*map*) dengan ordo 3×3 di dalam *game* yang akan dibangun.

Algoritma A* menggunakan dua senarai yaitu OPEN dan CLOSED. OPEN adalah senarai (*list*) yang digunakan untuk menyimpan simpul-simpul yang pernah dibangkitkan dan nilai heuristiknya telah dihitung tetapi belum terpilih sebagai simpul terbaik (*best node*) dengan kata lain, OPEN berisi simpul-simpul masih memiliki peluang untuk terpilih sebagai simpul terbaik, sedangkan CLOSED adalah senarai untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah terpilih sebagai simpul terbaik. Artinya, CLOSED berisi simpul-simpul yang tidak mungkin terpilih sebagai simpul terbaik (peluang untuk terpilih sudah tertutup). Setiap simpul (*node*) harus memiliki informasi nilai $h(n)$, yaitu estimasi harga simpul tersebut dihitung dari simpul tujuan yang hasilnya menjadi nilai H. Fungsi f sebagai estimasi fungsi evaluasi terhadap *node* n , dapat dituliskan :

$$f(n) = g(n) + h(n) \dots\dots\dots(1)$$

dengan :

$f(n)$ = fungsi evaluasi (jumlah $g(n)$ dengan $h(n)$)

$g(n)$ = biaya (*cost*) yang dikeluarkan dari keadaan awal sampai keadaan n

$h(n)$ = estimasi biaya untuk sampai pada suatu tujuan mulai dari n

Maka fungsi *heuristic* yang digunakan adalah sebagai berikut :

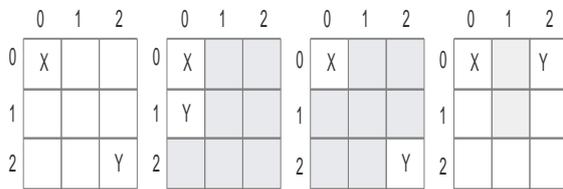
$$h_diagonal(n) = \min(\text{abs}(n.x-\text{goal}.x) + \text{abs}(n.y-\text{goal}.y))$$

$$h_orthogonal(n) = (\text{abs}(n.x-\text{goal}.x) + \text{abs}(n.y-\text{goal}.y))$$

$$h(n) = h_diagonal(n) + (h_orthogonal(n) - (2 * h_diagonal(n)))$$

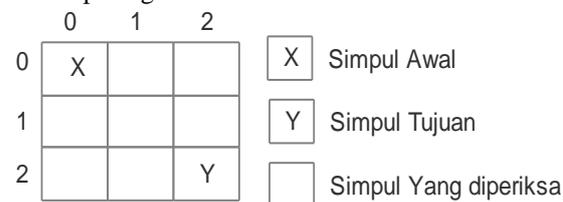
Sudah dijelaskan pada analisis masalah bahwa ordo dapat disesuaikan dengan rentang $X \times Y$, salah satu contoh perhitungan pada *game congklak* ini akan dijelaskan dengan ordo minimal yaitu 3×3 dikarenakan perhitungan pada ordo berapa pun akan sama. Pada penentuan bobot setiap *node* akan diberikan nilai sesuai dengan jarak terdekat ke tujuan, misal *node* yang terjauh dari tujuan maka diberi bobot yang kecil sedangkan *node* yang terdekat dengan tujuan diberi

bobot yang lebih besar. Contoh perhitungannya seperti pada gambar 3.4 yang diberi nilai bobot yang terkecil yaitu 1 dan bobot selanjutnya ditambahkan 1.



Gambar 2. Contoh kondisi ruang map yang akan dihitung dengan A*

Perhitungan yang dilakukan dengan algoritma A* dengan kondisi tanpa penghalang yang terlihat pada gambar 3.3.



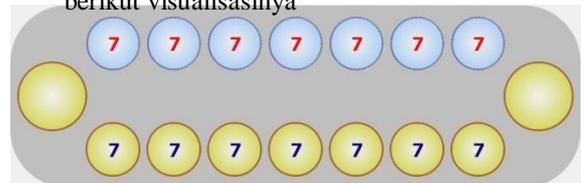
Gambar 3. Contoh kondisi tanpa penghalang dengan pencarian A*

Posisi simpul awal = Ax : 0, Ay : 0
 Posisi simpul tujuan = goal x : 2, goal y : 2
 Langkah ke satu
 $n(1,1) : g(1,1) = 1$
 $h_{orthogonal}(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))$
 $h_{orthogonal}(1,1) = (abs(1 - 2) + abs(1 - 2)) = (abs(-1) + abs(-1)) = 2$
 $h_{diagonal}(n) = min(abs(n.x-goal.x) + abs(n.y-goal.y))$
 $h_{diagonal}(1,1) = min(abs(1 - 2)+abs(1 - 2)) = min(abs(-1)+abs(-1)) = min 2$
 $h(n) = h_{diagonal}(n) + (h_{orthogonal}(n) - (2 * h_{diagonal}(n)))$
 $h(1,1) = (-2) + (2-(2*(-2))) = -2 + 6 = 4$
 $f(1,1) = g(1,1) + h(1,1) = 1 + 4 = 5$
 $n(1,0) : g(1,0) = 1$
 $h_{orthogonal}(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))$
 $h_{orthogonal}(1,0) = (abs(1 - 2) + abs(0 - 2)) = (abs(-1) + abs(-2)) = 3$
 $h_{diagonal}(n) = min(abs(n.x-goal.x), abs(n.y-goal.y))$
 $h_{diagonal}(1,0) = min(abs(1 - 2)+abs(0 - 2)) = min(abs(-1)+abs(-2)) = min 3$
 Proses diatas merupakan langkah yang harus dilakukan untuk menyelesaikan permainan

congklak, begitu proses seterusnya hingga semua biji congklak masuk kedalam lumbung besar, berikut adalah visualisasi permainan congklak dengan penyelesaian algoritma A* yang penulis rancang

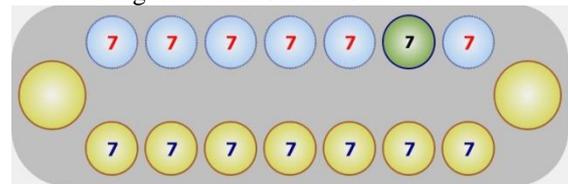
1. State Awal

Permainan congklak dengan penyelesaian algoritma A* memiliki state awal dimana semua posisi biji congklak memiliki nilai yang sama, berikut visualisasinya



Gambar 4. State Awal Congklak

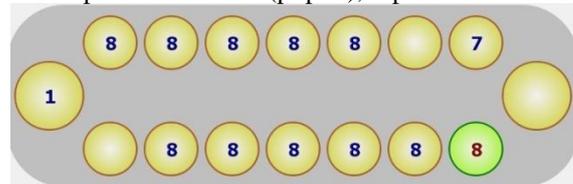
2. Setelah diketahui posisi awal permainan congklak dimana n=7 dan p=7, berikutnya diambil nilai dari posisi nilai manapun untuk dipindah ke lumbung kecil congklak, nilai pada lumbung kecil diambil random



Gambar 5. Pilihan Lumbung Kecil

Tampak pada gambar 5 pilihannya pada biji congklak nomor 2 dari sebelah kanan, penulis memberikan tanda warna yang berbeda agar mudah dikenali.

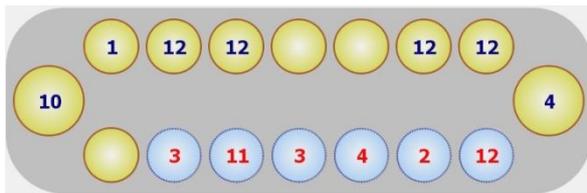
3. Berikutnya adalah jumlah biji n=7 akan dipindah ke posisi lumbung kecil yang berada di posisi kiri n=n-1 (p=p+1), seperti dibawah ini



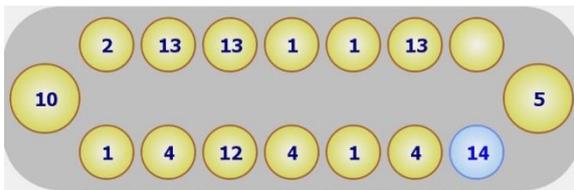
Gambar 6. Pergerakan Biji Congklak

Tampak pada gambar bahwa biji congklak sudah bergerak dari posisi awal ke posisi sebelah kiri dengan ditandai bawah biji setiap lumbung bertambah 1 dan posisi congklak awal sudah kosong, kemudian tampak pada gambar sebelah kiri posisi pertama sudah kosong yang berarti biji congklak berhenti dibagian tersebut dan maju ke sebelah kanan ditandai dengan jumlah biji bertambah 1 untuk posisi bawah.

4. Berikut adalah beberapa proses pemindahan yang terjadi dalam visualisasi permainan congklak dengan algoritma A*

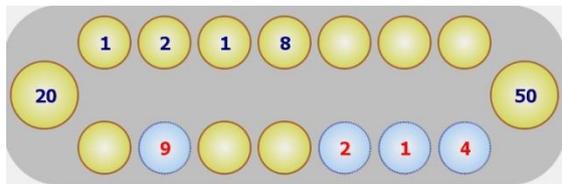


Gambar 7. Pergerakan Biji Congklak langkah ke-4



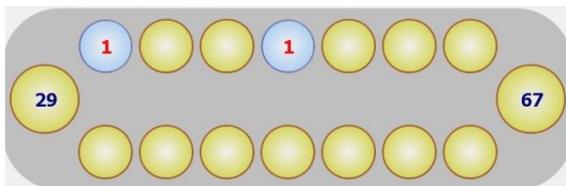
Gambar 8. Pergerakan Biji Congklak langkah ke-5

5. Proses pemindahan biji congklak ini dilakukan hingga semua posisi biji sudah pindah ke lumbung besar seperti dibawah ini



Gambar 9. Pergerakan Biji Congklak langkah ke-12

Berikut adalah tampilan permainan dalam kondisi berakhir



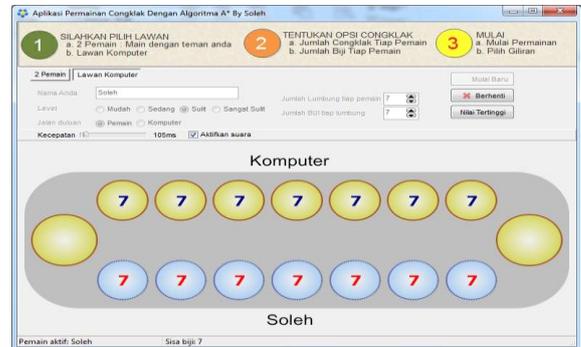
Gambar 10. Kondisi Selesai

Gambar 10 merupakan kondisi permainan nantinya jika diterapkan dengan menggunakan algoritma A* pasti akan ditemukan solusi hingga permainan berakhir, pada permainan yang dirancang terdapat fungsi algoritma A* yang bisa dijalankan otomatis untuk untuk bermain dengan menampilkan pergerakan secara dinamis.

IV. IMPLEMENTASI

Implementasi permainan dari game congklak yang penulis rancang, tampak nama pemain menggunakan nama penulis sendiri dan lawan

pemain adalah komputer, pada bagian komputer algoritma A* diterapkan dikarenakan jalannya komputer harus otomatis dan menghitung kemungkinan lebih tinggi untuk menang, pada langkah pertama penulis maju duluan



Gambar 11. Implementasi Program

V. KESIMPULAN

Kesimpulan dari penulisan penelitian ini adalah:

1. Solusi terbaik yang ditampilkan pada permainan congklak dengan menerapkan algoritma A* adalah dengan memilih biji congklak yang sesuai dengan jumlah yang diinginkan hingga lumbung congklak pada permainan penuh.
2. Permainan yang dirancang dengan menggunakan bahasa pemrograman Borland Delphi bisa menerapkan algoritma A* dengan baik dengan melakukan pergerakan yang paling baik
3. Cara kerja algoritma A* dilakukan secara langkah per langkah hingga permainan selesai dan pemain (komputer) menang dengan jumlah biji congklak terbanyak

DAFTAR PUSTAKA

1. T. Sutojo, Edy Mulyanto, Vincent Suhartono, "Kecerdasan Buatan", Penerbit Andi, Yogyakarta, 2011
2. Yulikus Partono, "Pengantar Logika Dan Algoritma", Penerbit Media Kita, Jakarta, 2008
3. Prabowo Pudjo Widodo, Herlawati, "Analisis Dan Perancangan", Penerbit Informatika, Bandung, 2011
4. Jogianto, Hartono "Analisis Dan Desain Sistem Informasi", Penerbit Andi, Yogyakarta, 2005
5. Intan Yuniar Purbasari, "Desain Dan Desain Algoritma", Penerbit Graha Ilmu, Surabaya, 2007
6. Eko Budi Purwanto, "Teori Dan Aplikasi Sistem Digital", Penerbit Graha Ilmu Yogyakarta, 2011
7. Abdul Kadir, "Pengenalan Sistem Informasi", Penerbit Andi, Yogyakarta, 2009