

Aplikasi Permainan Gomoku dengan Algoritma Negamax dan Alpha-Beta Search

Harris Kurniawan
STMIK Potensi Utama
E-mail: ch0c0_pahmen@yahoo.com

Abstrak

Permainan Gomoku adalah permainan dari jepang oleh dua orang yang dimainkan di atas papan Go. Go berarti lima, dan Moku adalah kata untuk batu yang berada di atas papan go dan narabe berarti berturut-turut. Pemain membuat aturan Renju agar permainan menjadi lebih variasi. Setiap pemain dapat mengetahui persis bagaimana posisi lawan dan pilihan langkah yang tersedia, berbeda halnya dengan permainan kartu, yang hanya dapat mengetahui kartu yang ada di tangan saja. Pemenangnya adalah pemain pertama yang mendapatkan 5 baris batu yang tidak terputus secara horizontal, vertikal, atau diagonal. Mencari pohon Negamax mengimplementasikan gagasan bahwa lebih buruk balasan lawan terbaik adalah, semakin baik Anda bergerak dan Alpa-Beta Search akan menganggap bahwa lawan adalah rasional ; yaitu , lawan bisa menghitung bergerak sebaik yang kita bisa, dan lawan akan selalu memilih langkah optimal dengan asumsi bahwa kita juga akan bermain dengan sempurna. Hasil dari penelitian ini adalah aplikasi permainan Gomoku yang dapat dimainkan pemain melawan komputer, juga komputer melawan komputer. Di mana daerah paling sudut di papan catur adalah daerah yang paling menguntungkan untuk setiap pemain. Berdasarkan hasil pengujian, didapatkan bahwa penggunaan metode Negamax dan dapat memberikan solusi pencarian langkah terbaik. Untuk mengetahui pemenang pada setiap permainan adalah dengan dengan jumlah biji hitam, maka pemenangnya adalah pemain dengan biji putih.

Kata Kunci — Gomoku, intelegensia buatan, Negamax dan Alpa-Beta Seach.

Abstract

Gomoku game is a game from Japan which is played by two men on the Go board. Go means five, and Moku is the word for a stone that is placed on the board and go narabe mean respectively. Renju players make the rules so that the game becomes more varied. Each player can know exactly the opponent position and move options available, unlike the card game, which can only know that the card is in the hand alone. The winner is the first player to get 5 stones unbroken line horizontally, vertically, or diagonally. The Negamax tree implements the idea that poorer opponent is, the better you move and Alpa - Beta Search will assume that the opponent is rational; ie, the opponent can calculate the best move we could, and the opponent will always choose the optimal step with the assumption that we will play perfectly. The results of this research is the application of Gomoku game that can be played against the computer players, also the computer against the computer. Where the area in the corner of the chess board is the most profitable areas for each player. Based on the test results, it was found that the use of the method and can provide solutions Negamax search the best move. To determine the winner in each game is with the number of black seeds, then the winner is the player with the white beans.

Keywords — Gomoku, artificial intelligence, Negamax and Alpa-Beta Seach.

1. PENDAHULUAN

Dunia komputer, tidak selalu berisi hal-hal yang serius dan berat, seperti yang selalu dibayangkan oleh orang pada umumnya yakni dunia permainan (game). Konsep permainan adalah memberikan rasa hiburan kepada pemain baik itu permainan yang dimainkan oleh satu orang atau lebih, bahkan ada beberapa permainan hasil kreasi orang dahulu merupakan wujud dari dunia sebenarnya sebagai contoh permainan Gomoku pada dasarnya menguji kecerdasan taktik seseorang di dalam mengatur biji Gomoku di dalam usaha mengalahkan lawan. Permainan Gomoku merupakan permainan yang menggunakan papan yang berkotak, menyerupai papan permainan catur, bedanya pada papan Gomoku tidak berpola hitam putih. Permainan ini memerlukan minimal dua orang untuk memainkannya dengan masing-masing memegang kendali warna hitam dan putih.

Gomoku adalah permainan papan strategi abstrak. Juga disebut Gobang, five in a row dan connect five yang berarti 5 berturut-turut. Gomoku secara tradisional dimainkan dengan batu Go (batu hitam dan putih) pada papan go (15x15 atau 19x19 perpotongan), batu juga tidak boleh dipindahkan atau dihapus dari papan, hanya boleh diletakkan saja. Gomoku juga dapat dimainkan dengan kertas dan pensil. Permainan ini dikenal di beberapa negara dengan nama yang berbeda. Hitam bermain pertama, dan pemain bergantian meletakkan batu pada titik yang kosong, tidak ada langkah terlarang seperti halnya Renju. Pemenangnya adalah pemain pertama yang mendapatkan 5 baris batu yang tidak terputus secara horizontal, vertikal, atau diagonal.

Nama "Gomoku" berasal dari bahasa Jepang. di mana ia disebut sebagai gomokunarabe. go berarti lima, dan Moku adalah kata untuk batu yang berada di atas papan go dan narabe berarti berturut-turut. Permainan ini juga populer di Korea, di mana ia disebut omok yang memiliki struktur asal nama yang sama seperti asal nama pada Jepang. Permainan ini dikatakan berasal di Cina dengan nama Wu Zi Qi. Pada abad kesembilan belas, permainan diperkenalkan ke Inggris di mana Gomoku dikenal sebagai Go Bang. Go Bang adalah kata yang merujuk pada Goban (bahasa Jepang) yang juga adalah k'i pan (qí bàn, bahasa Cina) yang berarti papan catur. Berdasarkan hal tersebut diatas maka masalah yang ada sebagai berikut:

1. Permainan Gomoku masih dimainkan secara manual.
2. Sering terjadinya kehilangan anak batu atau rusaknya papan permainan gomoku.
3. Pemain untuk menentukan langkah pada papan gomoku ditentukan dari lompatan dari setiap anak batu ke anak batu lawan agar menjadi miliknya.
4. Perpindahan objek pemain dilakukan secara manual dimana perpindahan dapat dilakukan secara horizontal, vertical dan diagonal.

Maka akan dibuat suatu sistem perangkat lunak permainan sederhana, dengan judul "Perancangan Perangkat Lunak Permainan Gomoku dengan Algoritma Negamax dan Alpha-Beta Search".

Sedangkan dari Jurnal Telematika Institut Teknologi Harapan Bangsa, Indonesia,2013 dengan judul "Implementasi Artificial Intelligence Pada Game Board Gomoku Dengan Menggunakan Aturan Renju" memakai metode Negamax.[2]

2. METODE PENELITIAN

- a. Analisis Kebutuhan sistem Perangkat Lunak Permainan Gomoku.
 - b. Spesifikasi dan Desain: Pada tahap ini dilakukan spesifikasi dan desain Aplikasi yang di bangun menggunakan bahasa pemrograman visual basis 6.0.
 - c. Implementasi dan Verifikasi : Pada tahap ini akan dilakukan implementasi dan verifikasi aplikasi, untuk menguji apakah aplikasi sudah berjalan sesuai dengan yang dirancang sesuai dengan manfaatnya
-

- d. Validasi yang penulis lakukan adalah melakukan pengujian sistem keseluruhan yang dirancang agar sistem yang dirancang sudah sesuai dengan aturan permainan gomoku, yaitu dapat melakukan perpindahan secara horizontal, vertical dan diagonal dan Pemain dapat menentukan langkah pada papan gomoku ditentukan dari lompatan dari setiap anak batu ke anak batu lawan agar menjadi miliknya.

2.1. Pencarian Negamax

Mencari pohon Negamax mengimplementasikan gagasan bahwa "lebih buruk balasan lawan terbaik adalah, semakin baik Anda bergerak."

Menerapkan ini sangat mudah. Ia menggunakan fakta bahwa catur adalah permainan simetris, dan karena itu fungsi analisis harus memberikan scoring simetris. Itu adalah untuk mengatakan bahwa pada titik tertentu, nilai untuk putih adalah persis dikurangi skor untuk hitam, atau ekuivalen jumlah dari dua nilai selalu sama dengan nol.

Hal ini cukup mudah untuk memahami. Jika putih menang dengan satu pion, maka jelas hitam kalah dengan jumlah yang sama. Prinsip ini dapat diperluas untuk keunggulan posisional, yaitu jika putih telah dua kali lipat Rooks pada satu file, maka putih memiliki skor bonus, sedangkan posisi hitam adalah lemah dengan jumlah yang sama karena hal ini.

Negamax hanya menerapkan langkah-langkah pencarian kasar berikut [5]:

```
Loop through all moves
  Play move
  move_score = - Opponent's_best_move_score
  if ( move_score > best_move_score ) then ( best_move =
    move )
  Undo Move
End of Loop
```

2.2. Pohon Permainan Alpha-Beta Search

Bagaimana kita menghitung langkah optimal kita? Kami akan menganggap bahwa lawan adalah rasional; yaitu, lawan bisa menghitung bergerak sebaik yang kita bisa, dan lawan akan selalu memilih langkah optimal dengan asumsi bahwa kita juga akan bermain dengan sempurna. (Kontras ini, misalnya, untuk pemain catur awal, yang sengaja akan membuat langkah dengan jebakan, dengan harapan menangkap lawan ke dalam perangkap dan mendapatkan kemenangan cepat. Namun, jika lawan tidak jatuh untuk perangkap, pemain kami menemukan bahwa posisinya kini kritis melemah).

Salah satu algoritma untuk menghitung langkah terbaik adalah algoritma minimax [5]:

```
minimax(player, board)
  if(game over in current board position)
    return winner
  children = all legal moves for player from this board
  if(max's turn)
    return maximal score of calling minimax on all the
      children
  else (min's turn)
    return minimal score of calling minimax on all the
      children
```

2.3. Analisis dan Perancangan

2.3.1. Analisis Masalah

Permainan Gomoku dilakukan pada sebuah papan berkotak dengan dimensi matriks 12 x 12 sebagai berikut:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|
| 1 | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | |

Gambar 1. Matriks 12 x 12

Permainan selalu dimulai dengan pemain yang memiliki biji warna hitam. Permainan Gomoku hanya dirancang untuk permainan maksimum 2 orang. Jenis permainan Gomoku untuk 2 orang pemain ataupun pemain 1 orang dengan komputer.

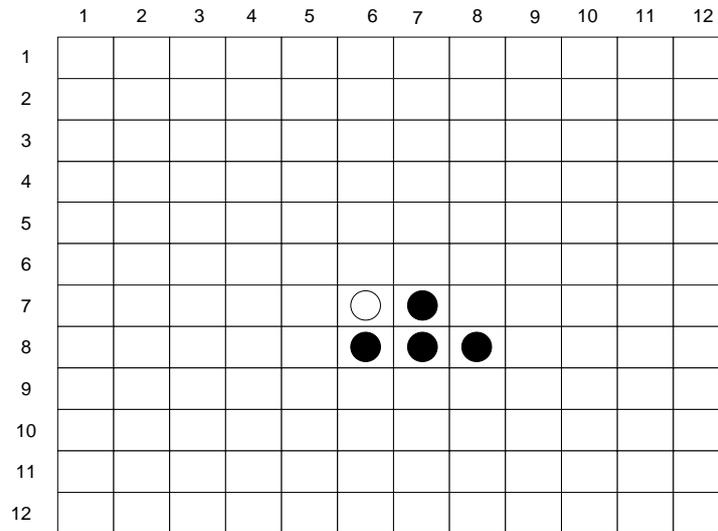
Aturan main baik untuk permainan 1 orang maupun permainan 2 orang adalah sebagai berikut:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|
| 1 | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | |
| 6 | | | | | | ○ | ● | | | | | |
| 7 | | | | | | ● | ○ | | | | | |
| 8 | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | |
| 10 | | | | | | | | | | | | |
| 11 | | | | | | | | | | | | |
| 12 | | | | | | | | | | | | |

Gambar 2. Awal Permainan

Misalkan pemain pertama adalah pemain dengan biji warna hitam dan pemain kedua adalah pemain dengan biji warna putih, awal permainan formasi di atas dengan 16x16 kotak.

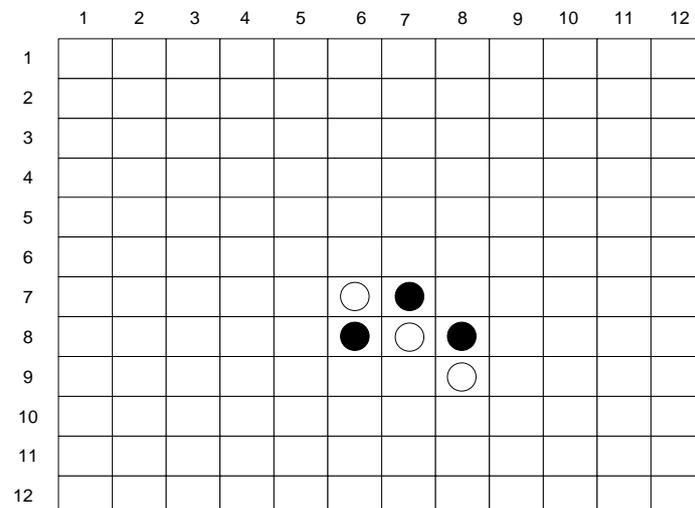
Pemain melangkah harus melompati biji pemain lawannya dan kotak tujuan adalah kotak kosong. Langkah boleh horizontal, vertical dan silang.



Gambar 3. Langkah Pemain Pertama

Pemain pertama seandainya melangkah ke 8-9 (baris 8 kolom 9) maka biji putih milik pemain pertama akan berubah menjadi warna hitam. Maka scorenya pemain pertama (hitam): 4 dan pemain (putih): 1.

Bila pemain kedua (putih) melakukan langkah ke 9-9 (baris 9 kolom 9) maka hasil permainan adalah sebagai berikut:



Gambar 4. Langkah Pemain Kedua

Demikian permainan seterusnya dilakukan hingga papan terisi biji penuh ataupun keduanya tidak dapat melangkah atau melanjutkan permainan lagi.

Untuk langkah komputer, diterapkan 2 metode, yakni:

1. Langkah Metode Negamax

Komputer akan melakukan perhitungan mencari daerah yang mungkin ditempati yang paling banyak. Komputer akan membandingkan langkah yang lebih baik, seperti : jika hasil perhitungan daerah pertama adalah 5 biji dan hasil perhitungan daerah kedua adalah 6 biji. Maka komputer akan mengambil langkah untuk daerah kedua. [4]

2. Langkah Alpha-Beta Search

Komputer akan melakukan perhitungan mencari daerah yang paling mendekati daerah sudut papan catur. Di mana daerah paling sudut di papan catur adalah daerah yang paling menguntungkan untuk setiap pemain. Sehingga dengan melakukan perhitungan, maka komputer akan selalu mengambil daerah yang paling mendekati daerah sudut papan catur.

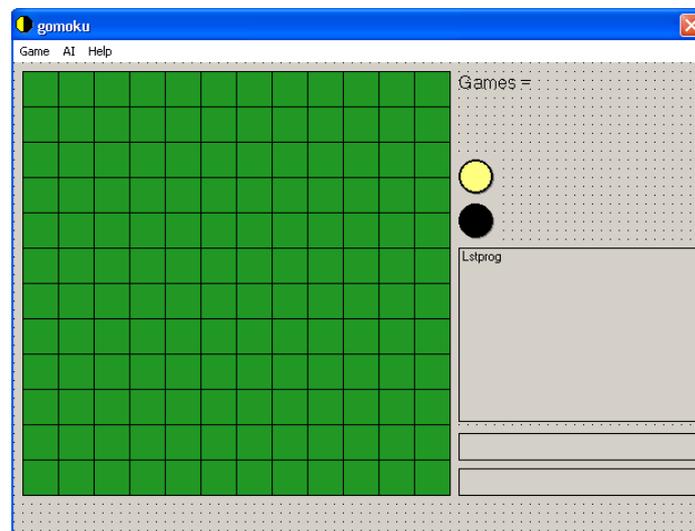
Untuk mengetahui pemenang untuk setiap permainan adalah dengan membandingkan jumlah biji putih dengan biji hitam. Jika jumlah biji putih lebih banyak dibandingkan dengan jumlah biji hitam, maka pemenangnya adalah pemain dengan biji putih. Demikian sebaliknya, jika jumlah biji hitam lebih banyak dibandingkan dengan jumlah biji putih, maka pemain dengan biji hitam adalah pemenang. Jika jumlah biji putih sama dengan jumlah biji hitam maka hasil permainan adalah seri. [4]

2.3.2. Perancangan

Setelah dilakukan analisa terhadap permainan gomoku, maka langkah selanjutnya dilakukan perancangan interface permainan. Untuk perancangan ini penulis menggunakan bahasa pemrograman visual basic 6.0 sebagai media untuk merancang papan permainan, koin dengan warna hitam dan putih serta lawan dalam permainan. Untuk aturan permainan yang terdapat pada permainan gomoku akan dituliskan ke dalam sintak program visual basic 6.0 dengan logika program sesuai dengan aturan pada visual basic 6.0. dengan demikian perancangan permainan gomoku nanti sesuai yang diinginkan oleh penulis.

2.3.2.1. Form Papan Gomoku

Perancangan papan Gomoku dapat dilihat pada gambar 5.



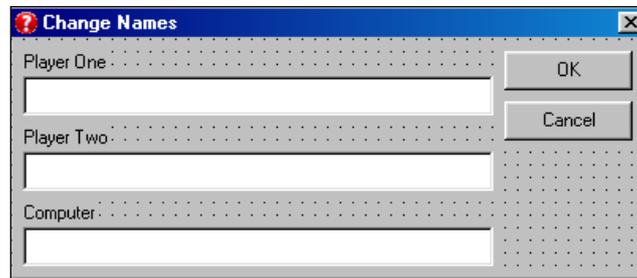
Gambar 5. Perancangan Form Papan Gomoku

Keterangan:

1. Menu : untuk menampilkan menu pilihan pada permainan gomoku.
2. PictureBox : Papan Gomoku, untuk menempatkan biji permainan Gomoku.
3. ImageList : untuk menyimpan gambar biji Gomoku.
4. label : Tempat untuk menampilkan informasi nilai (jumlah) biji Gomoku warna putih dan hitam.
5. lblmove : Tempat untuk menampilkan informasi tentang langkah-langkah pemain, apakah langkah tersebut adalah valid atau tidak.
6. lblstatus : Tempat untuk menampilkan status permainan.

2.3.2.2. Form Setting Nama Pemain

Adapun Form Setting Nama Pemain dapat dilihat pada gambar 6.



Gambar 6. Perancangan Form Setting Nama Pemain

Form ini digunakan untuk menggantikan nama dari setiap pemain, seperti pemain pertama, pemain kedua maupun nama dari komputer sebagai lawan main untuk jenis permainan tunggal.

2.3.2.3. Algoritam Program Permainan Gomoku

Pada algoritma permainan gomoku ini adalah merupakan aturan yang terdapat permainan gomokum nantinya program permainan ini akan sesuai dengan permainan yang dilakukan secara manual.

Berikut ini adalah algoritam permainan gomoku pada bahasa pemrograman visual basic:

1. Algoritma untuk memulai permainan gomoku [3].

```
Sub startup()  
  For outerloop = 1 To 12 'Clears the Grid Array  
    For innerloop = 1 To 12  
      gomokuGrid(outerloop, innerloop) = 0 'Clear grid  
    Next  
  Next  
  
  lblone.Caption = " 2 Counter(s)" 'updates the lbls  
  lbltwo.Caption = " 2 Counter(s)" 'updates the lbls  
  lblgames.Caption = Val(lblgames.Caption) + 1 'Displays games this  
session  
  SaveSetting App.EXEName, "PlayerID", "NumberOfGames",  
Val(lblgames.Caption)  
  If PlayerCounter = 1 Then 'Updates the counter player lbls  
    lblplayer.Caption = Playerone$ & " = Black"  
    If oneplayer = True Then  
      lblcom.Caption = Computer$ & " = White"  
    Else  
      lblcom.Caption = playertwo$ & " = White"  
    End If  
  Else  
    lblplayer.Caption = Playerone$ & " = White"  
    If oneplayer = True Then  
      lblcom.Caption = Computer$ & " = Black"  
    Else  
      lblcom.Caption = playertwo$ & " = Black"  
    End If  
  End If  
  Lstprog.Clear 'Clear computers captions  
  gomokuGrid(6, 6) = 1 'Add a White  
  gomokuGrid(7, 6) = 2 'Add a Black  
  gomokuGrid(6, 7) = 2 'Add a Black  
  gomokuGrid(7, 7) = 1 'Add a White
```

```

    For outerloop = 1 To 12 'This redraws the grid
      For innerloop = 1 To 12
        If gomokuGrid(outerloop, innerloop) = 0 Then
          LoadPicture(gomoku(((outerloop * 12) - 12) + innerloop - 1).Picture =
LoadPicture(""))
          ElseIf gomokuGrid(outerloop, innerloop) = 1 Then
            LoadPicture(gomoku(((outerloop * 12) - 12) + innerloop - 1).Picture =
white.Picture)
          ElseIf gomokuGrid(outerloop, innerloop) = 2 Then
            LoadPicture(gomoku(((outerloop * 12) - 12) + innerloop - 1).Picture =
black.Picture)
          End If
        Next
      Next
      Call message(" Status : " & Playerone$ & "'s Turn", True, False, False,
False, False) 'Change game status
      If currentuser = 1 And ShowCMove = 1 Then
        showmoves 'Display Possible Moves
      End If
    End Sub

```

2. Algoritma Pemilihan Kotak Pilihan [3].

```

Private Sub gomoku_click(Index As Integer) 'Pengguna memilih kotak pilihan
  Frmgomoku.MousePointer = 11
  Xposition = Int(Index / 12) + 1 'Capture A cords
  Yposition = Index - (Xposition * 12) + 13 'Capture B cords
  If gomokuGrid(Xposition, Yposition) = 0 Then
    Call GameEngine(Xposition, Yposition, PlayerCounter, 2) 'Call
placing sub
    Call message(Xposition & Yposition & squarepotent, False, False, False, False,
True)
    If goodshot = 1 Then
      Call message(" valid Move", False, True, False, False, False) 'A
valid move
      revert 'Revert Colours Back to Normal
      If oneplayer = True Then
        comp (3 - PlayerCounter) 'Compute computers go
        If ShowCMove = 1 Then
          showmoves 'Display Possible Moves
        End If
      Else
        PlayerCounter = 3 - PlayerCounter
      End If
    Else
      Call message(" Invalid Move", False, True, False, True, False)
'Not a valid move
    End If
  Else
    Call message(" Invalid Move", False, True, False, True, False)
'Not a valid move
  End If
  checkwin 'call sub to check if there is a winner
  Frmgomoku.MousePointer = 0
End Sub

```

3. Algoritma untuk perpindahan label pemain yang menentukan pilihan [3] :

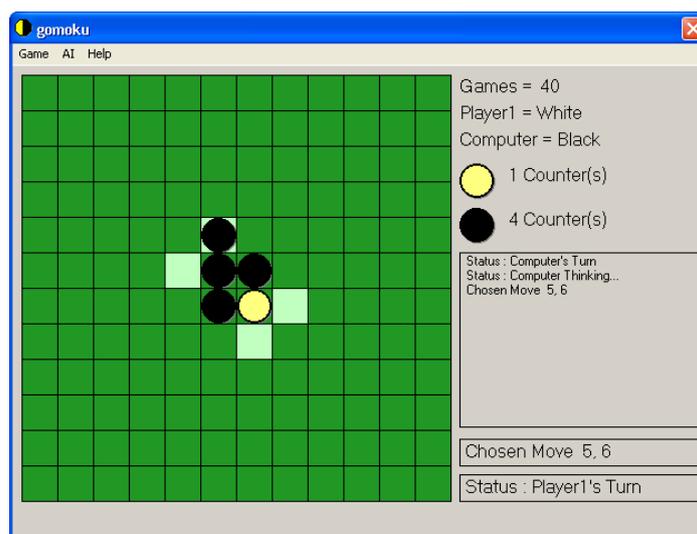
```
Sub showmoves()  
  For outerloop = 1 To 12  
    For innerloop = 1 To 12  
      SelectableSquares(outerloop, innerloop) = 0 'Clears the  
SelectableSquares array  
    Next  
  Next  
  For outerloop = 1 To 12  
    For innerloop = 1 To 12  
      If gomokuGrid(outerloop, innerloop) = 0 Then  
        Call GameEngine(outerloop, innerloop, currentuser, 1)  
'calls the gomoku game engine to check square  
      End If  
    Next  
  Next  
  For outerloop = 1 To 12  
    For innerloop = 1 To 12  
      If SelectableSquares(outerloop, innerloop) <> 0 Then  
        gomoku(((outerloop * 12) - 12) + innerloop - 1).BackColor =  
&HC0FFC0  
      End If  
    Next  
  Next  
End Sub
```

3. HASIL DAN PEMBAHASAN

3.1. Hasil

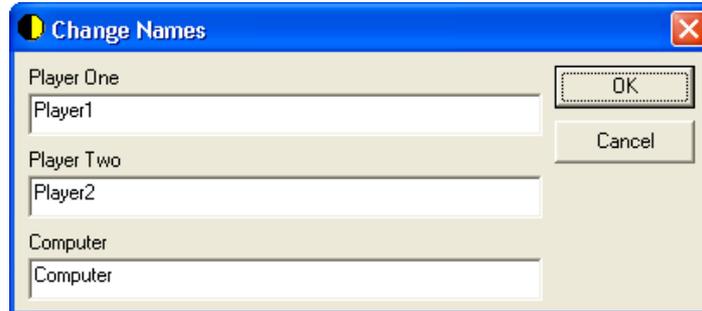
Pada permainan Gomoku original hanya menggunakan papan Gomoku berukuran 15x15. Untuk perancangan program pada penelitian ini, penulis melakukan modifikasi papan Gomoku dari ukuran 15x15 menjadi ukuran 12x12. Sedangkan untuk cara permainan, penulis tidak melakukan modifikasi.

Tampilan awal permainan Gomoku dapat dilihat pada gambar 7 berikut ini:



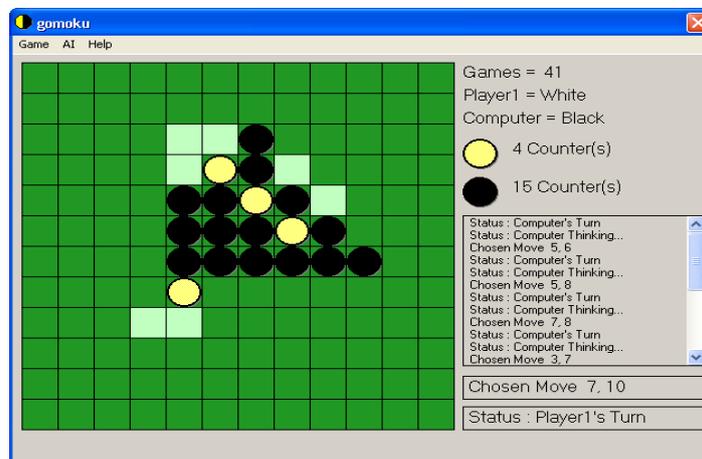
Gambar 7. Tampilan Awal Program [1]

Setelah itu pemain dapat mengisi nama pemain dan nama komputer seperti yang terlihat pada gambar 8 berikut ini:



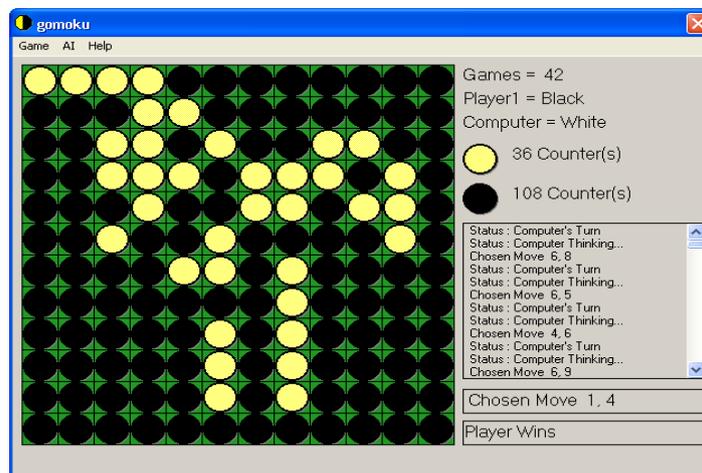
Gambar 8. Pengisian Nama Pemain [1]

Tampilan program ketika pemain telah meletakkan koin putih ataupun koin hitam dapat dilihat pada gambar 9 berikut ini:



Gambar 9. Pengisian Nama Pemain [1]

Tampilan program ketika papan Gomoku telah terisi semua dapat dilihat pada gambar 10 berikut ini:



Gambar 10. Tampilan Akhir Program

3.2. Analisa Hasil Program Gomoku

Tahapan perancangan aplikasi telah dikerjakan. Mulai dari rancangan sistem, dan perancangan antar muka (user interface). Semua rancangan ini digunakan untuk mempermudah dalam penjabaran sistem ke dalam bahasa pemrograman. Sebelum program diimplementasikan, maka program harus bebas dari kesalahan-kesalahan. Kesalahan program yang mungkin terjadi antara lain:

- a. Kesalahan penulisan bahasa (Syntax Error) adalah kesalahan dalam penulisan kode program. Dalam permainan Gomoku ini telah terbebas dari kesalahan tersebut, karena program permainan Gomoku ini dapat dijalankan sesuai dengan kebutuhan, seandainya terdapat kesalahan sudah bisa dipastikan program tidak bisa dijalankan.
- b. Kesalahan waktu proses (runtime-error) adalah kesalahan yang terjadi ketika executable program dijalankan, kesalahan ini akan menyebabkan program berhenti sebelum selesai pada waktunya, karena kompiler menemukan kondisi - kondisi yang belum terpenuhi yang tidak bisa dikerjakan. Dalam Program Permainan gomoku yang penulis rancang tidak terjadi erro sewaktu dijalankan dengan demikian program ini terbebas dari kesalahan waktu proses.
- c. Kesalahan logika (Logical errors) adalah kesalahan dari logika program yang dibuat. Kesalahan seperti ini sulit ditemukan, karena tidak ada pemberitahuan mengenai kesalahannya dan tetap akan didapatkan hasil dari proses program walaupun hasilnya salah. Setelah program bebas dari kesalahan, program tersebut diuji apakah sudah sesuai dengan tujuan atau belum. Dalam hal ini aturan dalam permainan gomoku sudah disesuaikan dengan aturan yang ada pada manual disesuaikan pada rancangan yang telah dibangun dalam hal ini belum ditemukan kesalahan logika.

3.3. Pembahasan

Dalam sistem ini penulis merancang sistem perangkat lunak permainan gomoku yang dapat dimainkan oleh 2 pemain. Program yang penulis buat cukup sederhana dan mudah untuk dipahami karena user hanya perlu mengklik tombol-tombol yang sudah tersedia sesuai kebutuhan. Alasan di atas dapat menjadi tujuan untuk meningkatkan Diperolehnya permainan Gomoku rancangan sendiri yang mudah dimengerti dan mudah untuk diimplementasikan serta dapat dimainkan oleh berbagai kalangan pengguna komputer. Memberikan suatu sarana hiburan permainan (game) untuk mengisi waktu-waktu luang. Dengan adanya aplikasi yang dihasilkan mampu memberikan permainan game yang dilakukan secara manual sekarang dapat dimainkan melalui komputer.

4. KESIMPULAN

Berdasarkan hasil analisa yang dilakukan, perangkat lunak yang telah penulis rancang masih jauh dari sempurna. Dari keseluruhan hasil pengujian yang dilakukan dapat disimpulkan sebagai berikut:

1. Penerapan metode Negamax dan Alpha-Beta Search jika hasil perhitungan daerah pertama adalah 5 biji dan hasil perhitungan daerah kedua adalah 6 biji dan Jika jumlah biji putih sama dengan jumlah biji hitam maka hasil permainan adalah seri.
 2. Pengembangan papan gomoku dari ukuran standard 15x15 menjadi 12x12 tidak mengakibatkan perubahan terhadap permainan. Cara pengambilan langkah adalah sama yakni horizontal, vertikal dan diagonal.
 3. Dengan menggunakan perangkat lunak, permainan Gomoku dapat menjadi lebih mudah dan praktis, di mana pemain tidak perlu membalik-balikkan biji lawan hasil pengambilan langkah yang termakan.
-

5. SARAN

Saran untuk penelitian lebih lanjut untuk menutup kekurangan penelitian dan dapat mengembangkan dari jurnal ini, boleh juga dengan menggunakan metode yang sama tetapi dengan membuat permainan games yang lain, atau bisa juga dengan games yang sama tetapi metode yang di gunakan berbeda untuk mendapatkan penelitian yang berbeda dari jurnal ini.

DAFTAR PUSTAKA

- [1] Sridadi, U. B., 2003, Pemodelan dan Simulasi Olah Yudha (War Game Simulation), *Jurnal Teknik UNJANI*, Vol 2, No 1, hal 128-135.
 - [2] Tanuwijaya, D., Implementasi Artificial Intelligence Pada Game Board Gomoku Dengan Menggunakan Aturan Renju, *Jurnal Telematika Institut Teknologi Harapan Bangsa*,
 - [3] Chess Tree Search, <http://www.cs.biu.ac.il/~davoudo/intro.html>, diakses tanggal 4 Januari 2014.
 - [4] Computer Chess Programming Theory, <http://www.frayn.net/beowulf/theory.html>, diakses tanggal 4 Januari 2014.
 - [5] Open Computing Facility at University of California, Barkeley
<http://www.ocf.berkeley.edu/~yosenl/extras/alphabeta/alphabeta.html>, diakses tanggal akses 5 Mei 2014.
-