

## PENGEMBANGAN APLIKASI PERMAINAN OTHELLO DENGAN NEGAMAX ALPHA BETA PRUNING DAN BRUTE FORCE

Siswanto<sup>1)</sup>, Laurensia<sup>2)</sup>, M. Anif<sup>3)</sup>

<sup>1, 2, 3)</sup> Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur  
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260

Telp. (021) 5853753, Fax. (021) 5866369

E-mail : [siswantobl@gmail.com](mailto:siswantobl@gmail.com)<sup>1)</sup>, [laurensiar@gmail.com](mailto:laurensiar@gmail.com)<sup>2)</sup>, [muhammad.anif@budiluhur.ac.id](mailto:muhammad.anif@budiluhur.ac.id)<sup>3)</sup>

---

### Abstrak

Kehidupan manusia yang semakin berkembang membuat perkembangan teknologi juga semakin pesat begitu pula dalam perkembangan dunia permainan yang juga berkembang pesat. Permainan yang ada digunakan tidak hanya untuk hiburan semata tetapi juga bisa untuk melatih menentukan strategi terbaik. Permainan tradisional yang dahulu dimainkan menggunakan alat-alat tradisional juga telah banyak dibuat ke dalam komputer dan sekarang juga merambah ke dunia *smartphone*. Othello adalah salah satu permainan tradisional yang dapat melatih kecerdasan untuk menyusun strategi terbaik dalam menentukan langkah - langkah permainan dengan tujuan untuk membalik keping lawan sebanyak – banyaknya, mencari posisi keping yang baik dan memenangkan permainan. Implementasi algoritma negamax alpha beta pruning dan brute force pada permainan othello berbasis android ini merupakan salah satu bentuk penerapan kecerdasan tiruan yang dapat membuat komputer menjadi lebih cerdas dan membuat permainan othello ini menjadi lebih menarik dan menghibur untuk dimainkan. Pengembangan aplikasi permainan othello pada platform android ini juga bertujuan untuk memudahkan pengguna karena lebih praktis, sehingga dapat dimainkan dimana saja dan kapan saja. Aplikasi permainan othello ini dikembangkan dengan menggunakan bahasa pemrograman Java dan berbasis Eclipse IDE. Hasil akhir yang diharapkan adalah sebuah aplikasi permainan othello dengan kecerdasan tiruan di dalamnya yang dapat diterapkan dalam *smartphone* android dengan sistem operasi 2.3.3 Gingerbread.

**Kata kunci:** permainan, negamax alpha beta pruning, brute force, android, othello

### Abstract

Human life is growing increasingly also make rapid technological developments as well as in the development of the game world are also growing rapidly. The game that is used not only for entertainment purposes only but can also determine the best strategy to train. The game is played using the traditional first traditional tools have also been made to the computer a lot and now also venturing into the world of *smartphones*. Othello is one of the traditional games that can train intelligence to devise the best strategy in determining the steps - steps to reverse the game with a goal as the puck opponents - many, finding a good puck position and win the game. Implementation negamax alpha beta pruning algorithm and brute force on the android based othello game is a form of artificial intelligence application that can make computers become more intelligent and make the othello game become more interesting and entertaining to play. Othello game application development on android platform also aims to facilitate users for a more practical, so it can be played anywhere and anytime. Othello game application is developed using the Java programming language and the Eclipse-based IDE. The end result is an application that is expected othello game with artificial intelligence in it that can be applied in a *smartphone* with android 2.3.3 Gingerbread operating system.

**Keywords:** game, negamax alpha beta pruning, brute-force, android, othello

### 1. PENDAHULUAN

Penggunaan *smartphone* sungguh memudahkan manusia dalam memperoleh informasi juga dalam mengerjakan pekerjaannya dan sebagai sarana hiburan yaitu permainan [1]. Permainan begitu digemari karena sangat menghibur dengan tantangan-tantangan tak terduga yang ada di dalamnya. Memainkan sebuah permainan dalam *smartphone* juga dirasa sangat mudah, cepat dan ringkas karena ukuran *smartphone* yang bisa dibawa kemana-mana [1]. Selain sebagai sarana hiburan, permainan juga dapat digunakan sebagai sarana untuk melatih strategi dan juga media hiburan [2]. Perkembangan permainan terutama di dunia *smartphone* terlihat sangat pesat, hal ini dapat dilihat dari banyaknya *developer* permainan yang bermunculan. Tidak dapat dipungkiri bahwa bermain

permainan itu menyenangkan oleh sebab itu banyak *developer* berlomba-lomba membuat permainan yang semakin baik dari hari ke hari [2]. Membuat permainan apalagi menanamkan sebuah kecerdasan tiruan (*Artificial Intelligence*) pada komputer terutama pada jenis permainan seperti permainan tradisional juga merupakan sebuah tantangan yang menarik karena penerapannya lebih sulit daripada mengajarkan seseorang untuk memainkannya secara manual. Tantangan lainnya adalah bagaimana membuat agar komputer melalui kecerdasan tiruan yang diterapkan dapat mengerti sebuah permainan secara keseluruhan. Salah satu permainan tradisional yang menarik adalah othello. Othello adalah permainan tradisional asal Jepang yang dimainkan oleh dua orang pemain dan menggunakan biji hitam dan putih [10]. Tujuan dari permainan ini adalah untuk membalik sebanyak-banyaknya biji milik lawan dengan cara mengepungnya. Algoritma *negamax alpha beta pruning* dan *brute force* digunakan untuk menentukan langkah terbaik pada permainan othello ini.

## 2. LANDASAN TEORI

Landasan teori yang digunakan dalam tulisan ini, meliputi:

### 2.1 Permainan Tradisional Othello

Permainan othello dimainkan oleh dua orang pemain di atas papan yang berukuran 8x8 dan menggunakan 64 biji yang memiliki dua sisi yaitu hitam dan putih. Aturan main dari othello adalah sebagai berikut [5]:

- Permainan dimulai dengan 2 biji hitam dan 2 biji putih berada di tengah papan permainan.
- Pemain menggerakkan biji permainannya bergantian, dengan biji hitam yang bergerak lebih dahulu.
- Pergerakan yang dimungkinkan adalah dengan meletakkan biji di kotak yang kosong yang berdekatan dengan biji milik lawan dan dapat membalik minimal satu atau lebih biji permainan milik lawan.
- Biji-biji lawan yang terapat diantara biji permainan milik kita harus dibalik. Pengapitan dapat dilakukan secara vertikal, horizontal atau diagonal tanpa kotak kosong diantaranya.
- Biji-biji permainan mungkin dibalik dari beberapa arah pada sekali pergerakan.
- Bila tidak ada langkah legal yang didapat seorang pemain maka pemain itu harus menunggu sampai langkah legal untuknya tersedia.
- Bila seorang pemain mempunyai paling sedikit satu langkah legal yang tersedia, ia harus membuat pergerakan dan tidak boleh melewatkan gilirannya.
- Permainan dilanjutkan sampai seluruh kotak pada papan terisi penuh atau tidak ada pemain yang memiliki langkah legal.

### 2.2 Kecerdasan Tiruan

“Kecerdasan tiruan atau *artificial intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat berpikir dan berperilaku serta melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia” [3]. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan jaman, maka peran komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu, komputer diharapkan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia, termasuk membuat permainan [4].

### 2.3 Algoritma Negamax Alpha Beta Pruning

Pencarian *negamax* merupakan bentuk lain dari pencarian *minimax* yang bergantung pada nilai *zero-sum* dari permainan dua pemain. Algoritma ini berdasarkan pada  $\max(a, b) = -\min(-a, -b)$  untuk menyederhanakan pelaksanaan algoritma *minimax* [7]. Lebih tepatnya, nilai posisi untuk pemain A dalam permainan tersebut adalah negasi dari nilai pemain B. Dengan demikian, pemain bergerak mencari langkah yang dapat memaksimalkan negasi dari nilai posisi yang dihasilkan dari langkah tersebut, posisi pengganti ini harus mengikuti nilai yang telah diberikan oleh lawan, maksudnya tidak peduli apakah A atau B adalah bergerak. Ini berarti bahwa prosedur tunggal dapat digunakan untuk menilai kedua posisi dan merupakan penyederhanaan atas *minimax*, yang mensyaratkan bahwa A memilih langkah dengan suksesor bernilai maksimum sementara B memilih langkah dengan suksesor bernilai minimum. Optimasi untuk algoritma *minimax* juga dapat berlaku untuk *negamax*. *Alpha-beta pruning* dapat mengurangi jumlah simpul algoritma *negamax* dalam mengevaluasi pohon pencarian dengan cara yang sama dengan penggunaannya dengan algoritma *minimax* [9]. Gambar 1 berikut adalah *pseudocode* untuk algoritma *negamax* dengan *alpha beta pruning* dengan ke dalam terbatas.

```

function negamax(node, depth,  $\alpha$ ,  $\beta$ , color)
  if depth = 0 or node is a terminal node
    return color * the heuristic value of node
  bestValue :=  $-\infty$ 
  childNodes := GenerateMoves(node)
  childNodes := OrderMoves(childNodes)
  foreach child in childNodes
    val := -negamax(child, depth - 1, - $\beta$ , - $\alpha$ , -color)
    bestValue := max( bestValue, val )
     $\alpha$  := max(  $\alpha$ , val );
  if  $\alpha \geq \beta$ 
    break
  return bestValue

```

Gambar 1. Pseudo-code Algoritma Negamax Alpha Beta Pruning [4]

## 2.4 Algoritma Brute Force

Algoritma *brute force* adalah sebuah pendekatan yang lempang (*straightforward*) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang melibatkan [6]. Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*). Algoritma ini memiliki moto yaitu: *Just Do It!*

Berikut ini adalah beberapa karakteristik algoritma *brute force*:

- Algoritma *brute force* umumnya tidak “cerdas” dan tidak mangkus, karena ia membutuhkan jumlah langkah yang besar dalam penyelesaiannya. Kadang-kadang algoritma *brute force* disebut juga algoritma naif (*naïve algorithm*).
- Algoritma *brute force* seringkali merupakan pilihan yang kurang disukai karena ketidakmangkusannya itu, tetapi dengan mencari pola-pola yang mendasar, keteraturan, atau trik-trik khusus, biasanya akan membantu kita menemukan algoritma yang lebih cerdas dan lebih mangkus.
- Untuk masalah yang ukurannya kecil, kesederhanaan *brute force* biasanya lebih diperhitungkan daripada ketidakmangkusannya. Algoritma *brute force* sering digunakan sebagai basis bila membandingkan beberapa alternatif algoritma yang mangkus.
- Meskipun *brute force* bukan merupakan teknik pemecahan masalah yang mangkus, namun teknik *brute force* dapat diterapkan pada sebagian besar masalah. Agak sukar menunjukkan masalah yang tidak dapat dipecahkan dengan teknik *brute force*. Bahkan ada masalah yang hanya dapat dipecahkan secara *brute force*. Beberapa pekerjaan mendasar di dalam komputer dilakukan secara *brute force*, seperti menghitung jumlah dari  $n$  buah bilangan, mencari elemen terbesar di dalam tabel, dan sebagainya.
- Selain itu, algoritma *brute force* seringkali lebih mudah diimplementasikan daripada algoritma yang lebih canggih, dan karena kesederhanaannya, kadang-kadang algoritma *brute force* dapat lebih mangkus (ditinjau dari segi implementasi).

## 2.5 Bahasa Pemrograman Java

Bahasa pemrograman Java pertama lahir dari *The Green Project*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan *Oak*. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, beserta sembilan *programmer* lainnya dari Sun Microsystem. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan [8].

## 2.6 Android

Android adalah salah satu sistem operasi yang berbasis Linux dan digunakan untuk telepon seluler seperti telepon pintar (smart phone) dan komputer tablet [1]. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak [2].

## 3. METODE PENELITIAN

Metode penelitian yang digunakan dalam pembuatan aplikasi permainan Othello adalah *Software Development Life Cycle (SDLC)* dengan tahapan(Fase), sebagai berikut:

### 3.1 Analisis Kelayakan

Peneliti telah melakukan analisis kelayakan termasuk analisis persyaratan aplikasi permainan othello dalam hal data input dan output yang diinginkan, proses yang diperlukan untuk mengubah input menjadi output. Analisis kelayakan juga mencakup kelayakan teknis aplikasi permainan Othello yang akan dibuat, dalam hal alat yang

tersedia perangkat lunak, perangkat keras, dan perangkat lunak profesional yang terampil. Pada akhir fase ini, dibuat laporan kelayakan untuk seluruh aplikasi permainan Othello.

### 3.2 Analisis kebutuhan dan spesifikasi

Peneliti telah melakukan analisis kebutuhan dan spesifikasi yang dibutuhkan aplikasi permainan Othello, termasuk mengumpulkan, menganalisis, memvalidasi, dan menetapkan persyaratan dari aplikasi permainan Othello.

### 3.3 Desain

Peneliti telah mendisain aplikasi permainan Othello berupa rancangan aplikasi, rancangan algoritma, rancangan layar aplikasi permainan Othello di *smartphone android* dengan sistem operasi 2.3.3 *Gingerbread*.

### 3.4 Coding

Desain aplikasi permainan Othello yang ditentukan dalam dokumen desain menggunakan bahasa pemrograman Java dan berbasis *Eclipse IDE* dieksekusi. Output dari fase ini adalah kode sumber untuk perangkat lunak yang bertindak sebagai masukan untuk pengujian dan pemeliharaan fase.

### 3.5 Pengujian

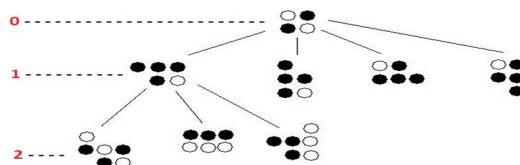
Proses pengujian aplikasi permainan Othello dimulai dengan rencana uji yang berulang, menggunakan uji kasus generasi, kriteria pengujian, dan alokasi sumber daya untuk pengujian. Kode ini diuji dan dipetakan terhadap dokumen desain dibuat dalam tahap desain. Output dari tahap pengujian adalah laporan pengujian mengandung kesalahan yang terjadi saat uji coba aplikasi, berupa kelebihan dan kekurangan program.

### 3.6 Pemeliharaan

Fase pemeliharaan juga mencakup penanganan kesalahan residual yang mungkin ada dalam perangkat lunak bahkan setelah fase pengujian.

## 4. RANCANGAN APLIKASI

Aplikasi permainan Othello yang akan dikembangkan menggunakan penerapan algoritma *brute force* dan *negamax alpha beta pruning* berbasis android. Pada aplikasi permainan ini terdiri dari dua mode yaitu 1 pemain dengan penggunaan algoritma *brute force* pada level mudah dan algoritma *negamax alpha beta pruning* pada level sedang dan sulit dan pada 2 pemain menerapkan level dengan perbedaan waktu di dalamnya dan juga mode *play to lose*. Algoritma *brute force* dan *negamax alpha beta pruning* digunakan untuk menentukan langkah terbaik yang dilakukan oleh komputer.



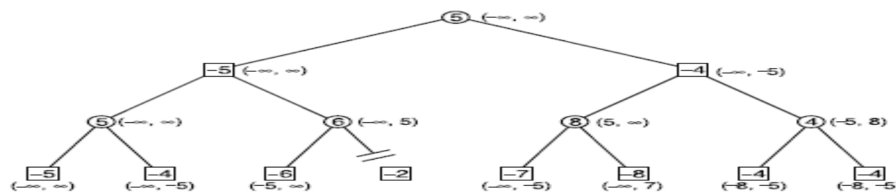
Gambar 2. Pohon Pencarian Othello

Algoritma *brute force* dan *negamax alpha beta pruning* akan melakukan pencarian dari tengah papan permainan, seperti yang diperlihatkan pada gambar 1. Kotak-kotak yang bukan merupakan langkah legal bagi komputer tidak akan ditelusuri lebih lanjut. Pencarian hanya dilakukan pada kotak-kotak yang merupakan langkah legal bagi komputer. Pencarian akan terus dilakukan sampai pada batas kedalaman yang telah ditentukan.

Algoritma *brute force* ini bekerja dengan cara sederhana, yaitu biji permainan akan diletakkan di kotak mana saja di papan permainan, asalkan langkah tersebut valid. Algoritma *brute force* ini tidak memperdulikan nilai kotak dari posisi yang diambilnya, yang penting biji permainan diletakkan di atas kotak yang valid. Dapat disimpulkan algoritma ini hanya mementingkan pemain dapat meletakkan biji permainan di atas papan permainan secepat mungkin.

Algoritma *negamax* mengimplementasikan bahwa semakin buruk langkah yang dilakukan oleh lawan artinya langkah yang kita lakukan semakin baik. Cara menerapkan algoritma *negamax* pada permainan othello adalah dengan menetapkan bahwa posisi nilai salah satu pemain adalah negasi dari langkah yang dilakukan oleh pemain lawan, atau bisa dibilang bahwa jumlah dari keduanya adalah 0. Apabila pemain unggul satu poin maka pemain lawan sudah pasti tertinggal sebanyak jumlah yang sama. Dasar dari algoritma ini adalah pergantian dari memaksimalkan dan meminimalisasi nilai dari posisi pada pohon pencarian. Selanjutnya, untuk membedakan posisi antara pemain dengan lawannya, nilai dari suatu posisi selalu dievaluasi dari sudut pandang pemain yang

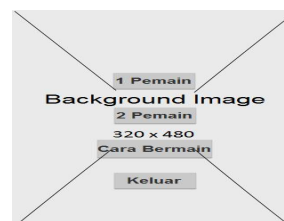
akan berjalan, hal ini dilakukan dengan melakukan negasi terhadap nilai yang dilihat oleh lawan, ini disebut dengan proses negamax. Dengan menggunakan algoritma *negamax* kita melakukan pencarian semua jawaban terhadap semua langkah dalam permainan. Karena begitu besarnya ruang pencarian maka algoritma *alpha beta pruning* dibutuhkan untuk mengurangi secara besar-besaran ukuran dari pohon pencarian.



Gambar 3. Pohon Pencarian Negamax Alpha Beta Pruning

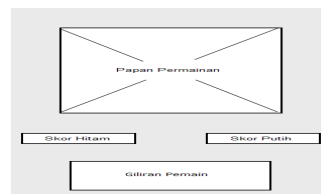
*Alpha beta pruning* sebagai algoritma pemangkasan akan menelusuri pohon pencarian pada permainan dengan meletakkan 2 nilai pada setiap *node*, yaitu *alpha* dan *beta* seperti gambar 3. Nilai *alpha* ditetapkan sama dengan  $-\infty$  (kondisi lawan memenangkan permainan) sedangkan nilai *beta* sama dengan  $+\infty$  (kondisi kita memenangkan permainan). Jika  $\alpha < \beta$ , maka kesempatan untuk mencari langkah terbaik masih ada dan pencarian akan tetap dilanjutkan. Selanjutnya *node* akan terus melakukan maksimalisasi dan memperbaiki nilai *alpha* dari *node* anak-anaknya, setelah itu nilai yang telah memperbaiki nilai *alpha* tersebut akan dibandingkan dengan *beta* sementara. Jika  $\alpha > \beta$  maka evaluasi akan dihentikan.

Pada aplikasi permainan othello ini terdapat rancangan layar menu utama seperti gambar 4 yang terdiri dari menu 1 pemain yaitu untuk bermain melawan komputer, menu 2 pemain yang digunakan untuk bermain dengan pemain lain, menu cara bermain untuk melihat cara bermain dan aturan permainan othello, menu keluar untuk keluar dari aplikasi permainan othello.



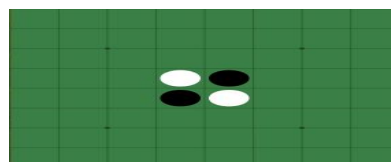
Gambar 4. Rancangan Layar Menu Utama

Pada saat bermain, pemain akan bermain pada papan berukuran 8 x 8 dan menggunakan 64 kotak seperti yang terlihat pada gambar 5.



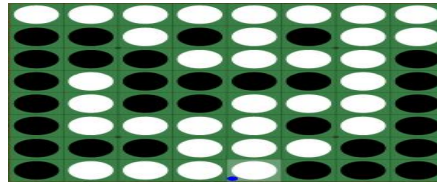
Gambar 5. Rancangan Layar Permainan Othello

Kondisi awal permainan Othello adalah 2 biji berwarna hitam dan 2 biji berwarna putih berada di tengah papan permainan seperti gambar 6. Dengan biji hitam merupakan biji permainan milik pemain dan biji putih merupakan biji permainan milik komputer. Pemain dan komputer hanya dapat menaruh biji permainan di kotak legal yang tersedia baginya. Pemain yang tidak memiliki langkah legal akan *pass* dan berganti giliran dengan pemain lainnya. Permainan terus berjalan dengan pemain dengan biji hitam dan biji putih saling berganti giliran.



Gambar 6. Kondisi Awal Papan Permainan Othello

Permainan berakhir apabila seluruh kotak pada papan permainan sudah terisi atau tidak ada lagi langkah legal yang tersedia bagi kedua pemain seperti gambar 7.



Gambar 7. Kondisi Pemain Putih Menang

Pemenang permainan Othello ditentukan berdasarkan banyaknya biji permainan yang terdapat pada pada permainan. Pemain yang menang adalah yang memiliki biji permainan terbanyak. Permainan berakhir seri apabila biji hitam dan biji putih memiliki jumlah yang sama.

## 5. HASIL DAN PEMBAHASAN

Implementasi aplikasi ini dibuat berdasarkan hasil analisa dan perancangan aplikasi permainan othello yang dikembangkan. Menu utama merupakan tampilan yang pertama kali muncul saat aplikasi dijalankan seperti gambar 8.



Gambar 8. Menu Utama Aplikasi

Ketika menu 1 pemain dipilih maka akan masuk ke menu pilih level yang terdiri dari 3 level yaitu mudah, sedang, dan sulit. Pada saat salah satu level dipilih maka akan muncul tampilan yang terdiri dari papan permainan, giliran permainan, dan skor dari masing-masing pemain.



Gambar 9. Tampilan Utama Permainan

Adapun kelebihan dan kekurangan aplikasi permainan othello ini, adalah sebagai berikut:

- a. Kelebihan Program
  - 1) Tampilan permainan yang menarik dengan animasi yang ada dalam aplikasi.
  - 2) Ditampilkannya langkah legal sehingga memudahkan pemain untuk memainkan *permainan* ini.
  - 3) Adanya *sound* pada saat menu diklik.
  - 4) Adanya fasilitas simpan skor sehingga pemain dapat menyimpan skor di akhir permainan.
  - 5) Adanya level pada mode permainan 2 pemain yang dibedakan berdasarkan waktu membuat permainan lebih menarik.
  - 6) Adanya fasilitas cara bermain sehingga pemain dapat melihat petunjuk cara bermain.
- b. Kekurangan Program
  - 1) Waktu berpikir yang cukup lama untuk komputer dengan level *hard*.
  - 2) Belum bisa dimainkan secara online atau *multiplayer* menggunakan *bluetooth*.
  - 3) Pemain belum bisa berganti biji permainan.
  - 4) Aplikasi permainan ini hanya terbatas pada layar android 320 x 480 pixel saja.
  - 5) Belum adanya *background* musik.

Spesifikasi hardware yang mendukung dalam pengoperasian aplikasi permainan othello yang dibuat, yaitu sebagai berikut: Tipe *input*: layar sentuh, Ukuran layar *display*: 320 x 480 pixel, RAM: 512 MB, Memori Internal : 512 MB dan *Single Core Processor* 1 GHz. Aplikasi permainan othello ini berjalan pada smartphone android dengan sistem operasi 2.3.3 Gingerbread.

## 6. SIMPULAN DAN SARAN

Simpulan dan saran yang dapat diambil dari tulisan ini, antara lain:

### 6.1. Simpulan

Simpulan yang dapat diambil dari tulisan ini, sebagai berikut:

- a. Aplikasi permainan othello dibuat untuk perangkat *mobile* berbasis android dengan sistem operasi 2.3.3 Gingerbread.
- b. Algoritma *negamax alpha-beta pruning* dan *brute force* dapat digunakan untuk menentukan langkah terbaik.
- c. Aplikasi permainan othello yang dikembangkan dapat digunakan sebagai sarana hiburan dan juga membantu untuk melatih pemain dalam menentukan strategi.

### 6.2. Saran

Saran untuk pengembangan selanjutnya, diharapkan aplikasi permainan othello yang dikembangkan memiliki tampilan yang menarik, sehingga dapat menarik minat pemain untuk memainkannya, pemain bisa berganti biji permainan, dapat jalan pada layar android di atas 320 x 480 pixel dan dilengkapi dengan background musik.

## 7. DAFTAR RUJUKAN

- [1] H., Nazruddin Safaat, 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Bandung: Informatika.
- [2] Handayani, Mauza Saputri, Dedy Arisandi, and Opim Salim Sitompul, 2012. *Rancangan Permainan Othello Berbasis Android Menggunakan Algoritma Depth-First Search*, Dunia Teknologi Informasi-Jurnal Online 1.1.
- [3] Siswanto, 2010. *Kecerdasan Tiruan, Cetakan Pertama, Edisi Kedua*, Yogyakarta: Graha Ilmu.
- [4] Millington, I. & Funge, J, 2009. *Artificial of Intelligence for Games*, Second Edition, Massachusetts: Morgan Kaufmann Publishers.
- [5] Rose, Brian, 2004. *Othello: A Minute to Learn A Lifetime to Master*, Connecticut: Anjar Co.
- [6] Munir, Rinaldi, 2004. *Bahan Kuliah ke-1 IF2251 Strategi Algoritmik*, Bandung: Departemen Teknik Informatika Institut Teknologi Bandung.
- [7] Ahmadi, Hamed, 2013, An Introduction to Game Tree Algorithms : *Negamax Algorithm (online)*, (update 26 May 2012 at 09:17), <http://www.hamedahmadi.com/gametree/#negamax> [Accessed 26 Juni 2014].
- [8] StackOverflow, 2011, *is-this-implementation-of-the-negamax-algorithm-correct in Java (online)*, (update 20 May 2011 at 11:06), <http://stackoverflow.com/questions/6070839/is-this-implementation-of-the-negamax-algorithm-correct> [Accessed 26 Mei 2014].
- [9] Tsan-sheng Hsu, 2012, *Alpha-Beta Pruning: Algorithm and Analysis (online)*, update 2012, <http://www.iis.sinica.edu.tw/~tshsu/tcg2012/slides/slide7.pdf> [Accessed 18 Mei 2014].
- [10] Argentahardi, Bina, Setyowati, Rizky Yuniar Hakkun, Yuliana, dan Rosyid Muhtadai, Nur, 2011, *Aplikasi Multiplayer Mobile Game Othello Dengan Menggunakan Bluetooth Sebagai Media Koneksi (Online)*, (update 4 Februari 2011 at 03:09), <http://repo.eepis-its.edu/605/1/1218.pdf> [Accessed 26 Juni 2014].